Estimation of Superimposed Convolutional Coded Signals

Gary Desmond Brushe

March 1996

A thesis submitted for the degree of Doctor of Philosophy of the Australian National University

Department of Systems Engineering
Research School of Information Sciences and Engineering
The Australian National University

Statement of Originality

The contents of this thesis is the result of original research and it has not been submitted for any other degree or award in any other university or educational institution.

G Brushe

Gary D. Brushe

6 March 1996

Acknowledgments

I would like to take this opportunity to thank the following people and organisations for their help during my doctorate:

- Dr. Langford B. White for his supervision, ideas, enthusiasm, in depth knowledge of signal processing and for introducing me to things I may not otherwise have been exposed to.
- Professor John B. Moore for his enthusiasm and supervision, especially during my residency at the Australian National University.
- The Cooperative Research Centre for Robust and Adaptive Systems (CRASys) without which this doctorate would not have been possible.
- The support of the Defence Science and Technology Organisation (DSTO), in particular, Communications Division, Mr. Neil Bryans and Dr. Ian Fuss for allowing me the opportunity to pursue this doctorate.
- Dr. Vikram Krishnamurthy, Dr. Mati Wax and Dr. Robert Mahony for their ideas, suggestions and interesting discussions.
- The members of the Signal Analysis Discipline and Digital Signal Processing Group, Communications Division, DSTO, and Ms. Joanna Spanjaard (CRASys) who are great people to work with, and with whom I have had many interesting and stimulating discussions.

- The staff and students from the Department of System Engineering and the staff from CRASys (located at ANU), who made my residency at ANU both enjoyable and rewarding.
- The South Australian Centre for Parallel Computing for use of the CM5 Connection Machine in the processing of some of the simulations.

I would like to acknowledge the funding of the activities of the Cooperative Research Centre for Robust and Adaptive Systems by the Australian Commonwealth Government under the Cooperative Research Centre Program.

I must also thank Ken and Veronica Day for their friendship and help, and Mr. Paul Malcolm for proof reading this thesis.

Finally, and most importantly, I would very much like to thank my wife, Mary-Anne and children, Mary and Angela, for all their love, support and understanding not only during my doctorate but always. I would like to also thank my parents Des and Margaret and Mary-Anne's parents Colin and Margaret for their love, help and support.

Abstract

This thesis examines a number of estimation problems involving digital communications signals, in particular convolutional coded signals. The following problems are considered:

- The estimation of the structure of a convolutional coded signal, and
- The estimation of superimposed convolutional coded signals.

The thesis also delves into the areas of maximum likelihood (ML) state sequence estimation and maximum *a posteriori* probability (MAP) state estimation, which can be achieved via the Viterbi algorithm (VA) and hidden Markov model (HMM) forward-backward algorithm (HFBA) respectively. This investigation resulted in the development of a Viterbi forward-backward algorithm (VFBA) and a hybrid Viterbi / HMM forward-backward algorithm (hybrid algorithm) which allows state estimates to be obtained by interpolating between the VA and HFBA.

The thesis covers areas of signal processing which include: HMMs; array processing; demodulation; reduced complexity processing; parameter estimation; and state estimation.

The contribution of the research in this thesis can be summarised as follows:

- The development of a technique for estimating the structure (i.e. the constraint length and generating polynomials) of rate $1/2$ convolutional coded signals from the received signal (i.e. the encoded binary data).

- The use of state sequence estimation combined jointly with parameter estimation in array processing problems, while also making use of the knowledge of the signal models. Joint estimation is achieved by iteratively estimating the state sequences and parameters of the signals. The method developed allows simpler arrays to be designed due to the threshold extension obtained. It also has the potential to increase the throughput of current Multiple Access channel systems, for example, satellite communications and digital mobile cellular phones, by using an antenna array. Simulations show that the joint estimator significantly improvements in the Bit Error Rate (BER) of the demodulated signals and the estimates of each signal's angle of arrival, when compared to a deterministic ML estimation method.

- The development of a reduced complexity VA (RCVA) which differs from known reduced state sequence estimators (RSSE) by allowing the number of reduced states to be varied during processing rather than remaining fixed. The performance of the RCVA is compared with a RSSE and the VA.

- Reduced complexity techniques for the joint estimation of superimposed convolutional coded signals. This includes using the expectation maximisation (EM) algorithm, the RCVA and the development of an on-line joint state sequence and parameter estimator, which includes using the RCVA and EM algorithm in on-line modes. The performance of the on-line joint estimator is compared with versions in which the RSSE and VA are used to estimated the state sequence. Simulations show that the estimator using the RCVA acquires lock onto the correct state sequence and the parameter estimates (on average) faster than the estimator using the standard RSSE technique, while only performing a little worse than the estimator using the full state VA.

- The development of a VFBA, which produces probability values for every state at each time (i.e. soft outputs), maximised over all state sequences passing through that state.
- The development of a hybrid algorithm which can interpolate between the MAP state estimates obtained by the HFBA and the ML state sequence estimate obtained by the VA. This interpolation is controlled by use of a variable parameter.

Preface

This thesis is divided into six chapters.

- Chapter 1 introduces the topics and problems which are considered, and provides some background information on the research areas of interest. It also details the thesis contribution and structure.

- Chapter 2 develops a technique for estimating the structure of a rate $1/Q$ convolutional coded signal from only the received encoded binary data.

- Chapter 3 details the joint state sequence and parameter estimator. It also presents a modification which reduces the computational complexity in the parameter estimation, by use of the EM algorithm. Simulations are presented to demonstrate the superior performance of these joint estimators over a traditional sequential estimator.

- Chapter 4 presents the development of a RCVA and on-line joint estimator. Simulations compare the performance of these with RSSE and VA versions.

- Chapter 5 examines the connection between ML state sequence estimation and MAP state estimation. This results in the development of a VFBA and a hybrid algorithm.

- Chapter 6 presents a summary of the thesis and conclusions which have resulted from the research conducted. It also presents areas in which further investigation could be conducted and some areas for future research.

The following is a list of papers which have been published in or submitted to refereed journals and conference proceedings. These papers are based on the research presented in this thesis. The conference papers contain material overlapping with the journal publications.

Journal Papers

- BRUSHE G.D., M. WAX AND L.B. WHITE (1995). "Determining the Constraint Length and Generating Polynomials of Rate $1/n$ Convolutional Coded Signals", *IEEE Signal Processing Letters*, vol. 2, no. 8, pp. 160 - 162.

- BRUSHE G.D. AND L.B. WHITE. "Spatial Filtering of Superimposed Convolutional Coded Signals", submitted for publication to *IEEE Transactions on Communications.*

- BRUSHE G.D., V. KRISHNAMURTHY AND L.B. WHITE. "A Reduced Complexity On-Line State Sequence and Parameter Estimator for Superimposed Convolutional Coded Signals", submitted for publication to *IEEE Transactions on Communications.*

- BRUSHE G.D., R.E. MAHONY AND J.B. MOORE. "A Soft Output Hybrid Algorithm for ML / MAP Sequence Estimation", submitted for publication to *IEEE Transactions on Information Theory.*

Conference Papers

- BRUSHE G.D. AND L.B. WHITE (1995). "Joint Parameter Estimation and Demodulation of Superimposed Convolutional Coded Signals", in *Proceedings ICASSP*, Detroit, vol. 3, pp. 1788 - 1791.

- BRUSHE G.D., R.E. MAHONY AND J.B. MOORE. "A Forward Backward Algorithm for ML State and Sequence Estimation", submitted to the 1996 *International Symposium on Signal Processing and its Applications*.

- MAHONY R.E., G.D. BRUSHE AND J.B. MOORE. "An Investigation of the Relationship between ML and MAP Sequence Estimation Algorithms", submitted to the 1996 *International Symposium on Signal Processing and its Applications*.

Contents

List of Figures

Abbreviations

AMPP	*A Posteriori* Maximum Path Probability.
AOA	Angle Of Arrival.
AR	Auto-Regressive.
BER	Bit Error Rate.
CC	Convolutional Coded.
CDMA	Code Division Multiple Access.
dB	deciBel.
DF	Direction Finding.
DSP	Digital Signal Processing.
EM	Expectation Maximisation.
Eqn	Equation number.
E-step	Expectation step.
FDMA	Frequency Division Multiple Access.
FIM	Fisher Information Matrix.
GF	Galois Field.
GSM	Global System for Mobile communications.
HFBA	HMM Forward Backward Algorithm.
HMM	Hidden Markov Model.
ICASSP	International Conference on Acoustics, Speech and Signal Processing.
IEEE	Institution of Electrical and Electronic Engineers.
i.i.d.	Independently and Identically Distributed.
ISI	Inter Symbol Interference.

LSB	Least Significant Bit.
MA	Multiple Access.
MAP	Maximum *A Posteriori* Probability.
ML	Maximum Likelihood.
MSB	Most Significant Bit.
M-step	Maximisation step.
MUSIC	MUltiple Signal Classification.
NCDMC	Nearly Completely Decomposable Markov Chain.
PDMA	Polarisation Division Multiple Access.
QPSK	Quadrature Phase Shift Keyed.
RCVA	Reduced Complexity Viterbi Algorithm.
RMS	Root Mean Square.
RSSE	Reduced State Sequence Estimation.
SDMA	Space Division Multiple Access.
SKMA	Segmental K-Means Algorithm.
SNR	Signal to Noise Ration.
TDMA	Time Division Multiple Access.
ULA	Uniform Linear Array.
VA	Viterbi Algorithm.
VFBA	Viterbi Forward Backward Algorithm.
WGN	White Gaussian Noise.

Glossary

A	Transition probability matrix.
$\mathbf{A}(\Omega(t))$	Array steering matrix.
a_{ij}	Transition probability, state i to j.
B	Observation symbol probability matrix.
B	Hankel Matrix of message bits.
$b_j(U(t))$	Probability of observing state j given observation U(t).
b(t)	Message bits.
$C(q_t)$	Modulation function.
c(k)	Bits of Convolutional Code's shift register.
d	ULA sensor spacing.
d(k)	Bits of Convolutional Code's shift register.
$d_j(\mu)$	Continuous function mapping $\Re^+$ to $\Re^+$.
F	No. of states.
G	Toeplitz Matrix of generating polynomial(s).
$G_m^{(\ell)}(n)$	m^{th} generating polynomial for ℓ^{th} signal.
$\mathbf{g}^{(i)}$	i^{th} generating polynomial.
$g_j^{(i)}$	j^{th} coefficient of i^{th} generating polynomial.
H	No. of intervals in a search grid.
h	Delay after which a state doesn't affect an observation.
$I_\ell(t)$	Defined using the FIM or score vector of $Q_\ell(t)$.
I	Identity Matrix.
i	Index counter.
j	Index counter.
K	No. of sensors in ULA, also Sliding window size.
k	Index counter - no. of sensors.
$L(\Theta)$	Log-likelihood function.
L	No. of signals.
ℓ	Index counter - no. of signals.
M	No. of phase signalling values.
N(t)	Additive WGN matrix.
N	Constraint Length.
n(t)	Additive noise.
O(x)	Function of order x.

$P(\Omega)$	Orthogonal projection matrix spanning columns of $\mathbf{A}(\Omega(t))$.	$u_t^{(i)}$	Encoded bits due to i^{th} generating polynomial.
p(t), P(t)	Partition of s(t), S(t).	v(t)	Additive WGN.
Q	No. of generating polynomials.	$\mathbf{W}_j^{(i)}$	Toeplitz matrix of left null-vector $\mathbf{w}_j^{(i)}$.
$Q_\ell(t)$	Log-likelihood function.		
q_t	State at time t.	$\mathbf{W}_j$, $\mathbf{W}$	Matrices of $\mathbf{W}_j^{(i)}$.
$\Re^+$	All non-negative reals numbers.	$\mathbf{w}_j$, $\mathbf{w}_j^{(i)}$	Left null-vector(s).
		$w_{jk}^{(i)}$	k^{th} coefficient of $\mathbf{w}_j^{(i)}$.
$\hat{\mathbf{R}}$	Sample covariance matrix.	w(t), w_t	Additive WGN.
$\boldsymbol{R}(t)$	Noise covariance matrix.	x(t)	Real component of transmitted signal.
r(t), R(t)	Partition of s(t), S(t).		
s(t), S(t),		y(t)	Quadrature component of transmitted signal.
S_i	State of Convolutional Code(s) shift register.	Z	Message bits coded at one time.
T	Block length of data.		
t	Index counter - time.	$z(s(t),\Theta(t))$,	
U	Hankel matrix of encoded bits.	z(t), Z(t)	Complex transmitted signal(s).
u(t), U(t),		z^{-j}	Delay operator.
U_t	Observation(s).		

$\alpha_t(i)$	HMM's forward information probability measure.	$\gamma_\ell(t)$	Represents each parameter of the ℓ^{th} signal.
$\beta_t(i)$	HMM's backward information probability measure.	$\gamma_t(i)$	The HMM's *a posteriori* information probability measure for the i^{th} state.
$\tilde{\beta}_t(i)$	VFBA's backward path information probability measure.	$\tilde{\gamma}_t(i)$	The VFBA's AMPP measure for the i^{th} state.

$\delta_t(i)$	VA and VFBA's forward path information probability measure.
ζ_ℓ	Arbitrary real-valued scalars, that sum to 1.
Θ, $\Theta(t)$	Parameter vector.
θ, $\theta(t)$	AOA's azimuth of signal.
$\theta_t^\mu(i)$	Hybrid algorithm's *a posteriori* information probability measure.
$\kappa_t^\mu(i)$	Hybrid algorithm's forward information probability measure.
λ	HMM parameter set.
λ	Signal wavelength.
μ	Reduced memory of Convolutional Code, also a positive real parameter used in the hybrid algorithm.
ξ	Convergence criterion threshold.
π	Initial state distribution probability vector.
π	Pi (= $^{22}/_7$).
π_i, $\pi(i)$	Probability of initially being in state i.
P	Amplitude vector of signals
ρ, $\rho(t)$	Amplitude of signal.
σ	Noise standard deviation.
σ^2	Noise variance.
$\tau_t^\mu(i)$	Hybrid algorithm's backward information probability measure.
φ, $\varphi(t)$	AOA's elevation of signal.
Ψ	Phase offset vector of signals.
ψ, $\psi(t)$	Phase offset of signal.
$\psi_t(i)$	Matrix for VA backtracking.
Ω, $\Omega(t)$	AOA vector of signals.

Chapter 1

Introduction

The estimation of superimposed signals occurs in a variety of electromagnetic radiation systems, including: communications, medical imaging and radar. The detection and estimation of superimposed signals was studied by Wax in 1985. His dissertation "addresses the problem of estimating the number, the parameters and the waveforms of superimposed signals, occurring in a variety of fields ranging from radar, sonar, oceanography and seismology to medical imaging and radio-astronomy" [Wax 1985].

Wax traced the history of the superimposed signals problem from its possible beginnings in 1795, when Gaspard Riche, Baron de Prony published his work on fitting superimposed exponentials to data.

This dissertation examines the problem of digital communication signals, in particular, convolutional coded signals (described in Section 1.2). The transmission of digital data has become prolific in the last decade or so, and continues to grow rapidly as digital signal processing (DSP) chips become smaller, faster and more affordable. Convolutional codes are very popular in digital communications systems (e.g. satellite communications [Intelsat 1987] and more recently digital mobile cellular phone [Padgett *et al.* 1995, Steele 1992] systems) because of their optimal decodability by the efficient Viterbi algorithm (VA) [Viterbi 1967, Forney 1973]. A brief description of the VA is provided in Appendix A.

Communications systems use Multiple Access (MA) schemes to allow multiple signals to be received apparently simultaneously. These MA schemes include: Time Division MA (TDMA); Frequency Division MA (FDMA); Code Division MA (CDMA); Space Division MA (SDMA); and Polarisation Division MA (PDMA). However, only one signal can be received in each division at any one time, i.e. in each time division for TDMA, in each frequency division for FDMA, using each (nearly) orthogonal code for CDMA, by directing spot beam antennas at each signal for SDMA, and in each polarisation for PDMA. Using an array of sensors, a method is developed which can accurately demodulate superimposed convolutional coded signals and thus could be used to increase the throughput of digital communications systems which use these MA schemes. Array processing techniques have been used to spatially filter superimposed signals before individually demodulating them [Haykin 1991]. Currently, spatial filtering of signals requires the estimation of each signal's angle of arrival (AOA) using direction finding (DF) techniques [Hurt 1990]. These techniques use no knowledge of the models of the signals being received. The method developed in this dissertation improves the accuracy (when compared to deterministic maximum likelihood (ML) methods, e.g. the method by Wax [1985], which is briefly described in Appendix E) of estimating superimposed signals, by using knowledge of the signal models. The method jointly estimates each signal's parameters (e.g. AOA) and demodulates them. Suboptimal methods which reduce the computational complexity of the above joint estimation technique are also developed. These include: the use of the expectation maximisation (EM) algorithm for superimposed deterministic signals (which is briefly described in Appendix D); the development of a reduced complexity Viterbi algorithm (RCVA); and the development of a reduced complexity on-line joint estimator using both the RCVA and EM algorithms in on-line modes.

In order to decode a convolutional coded signal the VA requires exact knowledge of the structure of the convolutional encoder. A method which determines the convolutional encoder's structure from the received signal is developed in this dissertation.

The VA maximises a forward path probability (see Appendix A) in order to estimate a ML state sequence, via a backtracking procedure, from noisy observations. In an analogous manner a backward path probability is generated, replacing of the backtracking procedure. This lead to the development of a Viterbi forward-backward algorithm (VFBA). This algorithm computes an *a posteriori* maximum path probability (AMPP) for each state at each time thereby providing a confidence level for each state estimate. The similarity of the VFBA's structure with that of the hidden Markov models (HMMs) forward-backward algorithm (HFBA), which is used for Maximum *a posteriori* Probability (MAP) state estimation, is exploited to develop a hybrid algorithm. A description of the hidden Markov model is given in Section 1.3 and the HFBA in Appendix B. The hybrid algorithm provides a method to adaptively interpolate between obtaining the ML path estimate and MAP state estimate from noisy observations of a Markovian state sequence.

The rest of this chapter provides brief introductions on a number of topics which provide background information to this dissertation. The topics are: a communications system; convolutional coded signals; the hidden Markov model (HMM); and array processing. Known algorithms which are used in this dissertation are briefly described in Appendices A - E with references to material which can provide the reader with more detailed explanations. The algorithms are respectively: the Viterbi algorithm (VA); the HMM forward-backward algorithm (HFBA); the segmental k-means algorithm (SKMA); the expectation maximisation (EM) algorithm for superimposed deterministic signals; and direction finding via a method of maximum likelihood.

At the end of this chapter the algorithms developed in this thesis are summarised and an outline of the remaining thesis structure is presented.

§1.1 Communications System

A communication system's function is to transmit information from a source to a destination via a channel using some carrier signal. The basic communication system consists of the following components.

Source —> Transmitter —> Channel —> Receiver —> Destination.

In this dissertation only a digital system is considered. In a basic digital system, on the source side the information is formatted and modulated before being transmitted into the channel, however in more complicated systems it may also contain encoding (source and channel), encryption, multiplexing, frequency spreading and multiple access components. At the receiver there needs to be the corresponding inverse components in order to recover the original message. The reader is referred to any digital communications text book for more information, e.g. Sklar [1988] or Haykin [1988].

The only extra component added to the basic digital system, described above, (that is considered in this dissertation) is that the information bits are convolutionally encoded prior to transmission. It is assumed that: the channel only adds white Gaussian noise (WGN) to the signal(s); does not impose any inter symbol interference (ISI) on the signal(s); is void of multipath; the receiver samples the signal(s) once per baud; when multiple signals are being sampled, the signals have the same baud rate; and the signal(s) have been mixed down to baseband.

The ISI and equalisation problems in the reception of digital communications signals are not considered in this dissertation as they are outside the scope of this work. Eyuboğlu and Qureshi [1988], Duel-Hallen and Heegard [1989], Hagenhauer and Hoeher [1989], Tong *et al.* [1991 and 1993], Moulines *et al.* [1994] and Xu *et al.* [1994], are just a few of the researchers who are investigating these problems.

§1.2 Convolutional Coded Signals

Convolutional codes were first proposed by Elias in 1955. A convolutional coded signal has digital input information bits (the message sequence) $b(t), t \geq 0$ which can be denoted as a first order Markov process. If the $b(t)$s are an independently and identically distributed (i.i.d.) equiprobable binary process, then the transition probabilities , a_{ij}† $= 0.5$ for $i, j = 0,1$. The message sequence can be convolutionally encoded via a linear operation (over GF(2), i.e. modulo-2, [Blahut 1984]) using N-1 shift registers and Q generating polynomials to produce a constraint length N and rate Z/Q (where Q coded bits are produced for every Z information bits) convolutional coded bit stream. The coded bit stream is then transmitted using a modulation scheme. If M-ary phase shift keyed modulation [Sklar 1988] is used, the coded bits determine one of $M = 2^Q$ possible phase signalling values. The phase value, $\phi(t)$, of the signal is thus determined by:

$$\phi(t) = \frac{2\pi}{M}\sum_{m=0}^{M-1} 2^m \sum_{n=0}^{N-1} b(t-n)G_m(n), \tag{1.1}$$

where $G_m(n)$ are known binary coefficients of the convolutional code's generating polynomials.

Figure 1.1 shows how to generate a rate $1/2$, constraint length 7, convolutional coded quadrature (i.e. M = 4) phase shift keyed (QPSK) signal‡ from the binary input information bits, $b(t)$ using the generating polynomials $G_0(n)$ and $G_1(n)$.

Combining convolutional coding and QPSK modulation as in Eqn. (1.1) is done so as to reflect the way it is done in the possible application areas. Some satellite communication systems (e.g. Intelsat [1987]) use punctured QPSK convolutional coded signals. Punctured codes can be decoded as unpunctured codes with minimal

† a_{ij} is the transition probability from state i to state j.

‡ The theory presented in this thesis is not restricted to just QPSK signals, any digital modulation type may be used.

increase in complexity [Bibb Cain *et al.* 1979]. Digital mobile phones (e.g. GSM) [Padgett *et al.* 1995, Steele 1992] use convolutional coded signals with QPSK modulation or its variants [Falconer *et al.* 1995].

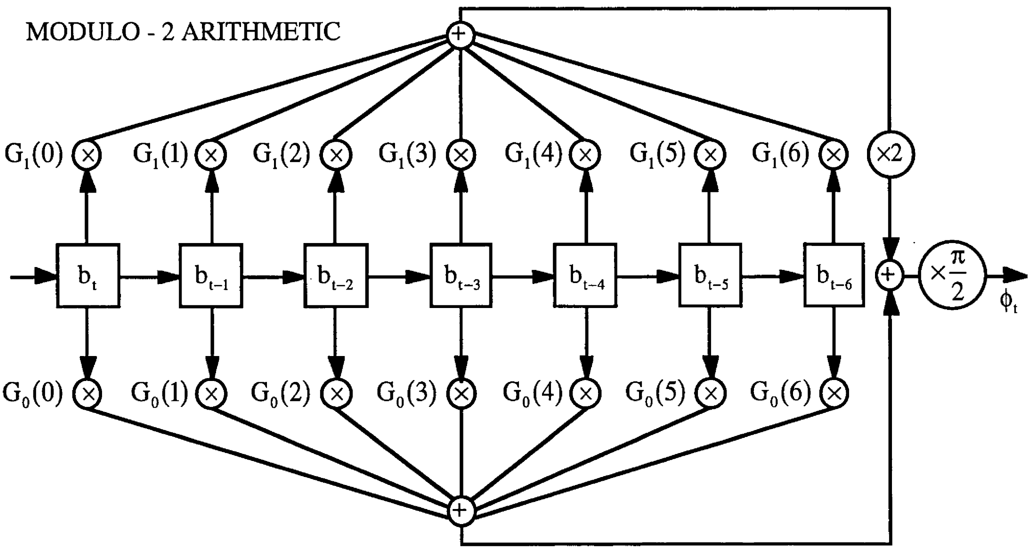

Figure 1.1: Generation of a Convolutional Coded QPSK Signal.

Let

$$s(t) = [b(t-N+1),\ldots,b(t-1),b(t)] \tag{1.2}$$

be the state of the convolutional code's shift register at time t. The state sequence $\{s(t)\}$ is a first order Markov process having $F = 2^N$ states with transition probabilities:

$$\Pr\{s(t) = [c(N-1),\ldots,c(0)] \mid s(t-1) = [d(N-1),\ldots,d(0)]\} = \begin{cases} a_{d(0)c(0)} & c(j) = d(j-1), \quad 1 \le j \le N-1 \\ 0 & \text{else} \end{cases} \tag{1.3}$$

where the c(j)s and d(j)s are dummy variables used to represent the elements of s(t) and s(t-1) respectively.

In this thesis attention is focused on the uniform i.i.d. message case, although such a restriction is not necessary for any of the analysis presented herein to remain valid. Each $s(t)$ corresponds to a certain $\phi(t)$, hence the transmitted signal is modelled by

$$z(s(t),\Theta(t)) = \rho(t)e^{i(\phi(t)+\psi(t))} = x(t) + i\, y(t) \quad \text{with} \quad \begin{aligned} x(t) &= \rho(t)\cos(\phi(t)+\psi(t)) \\ y(t) &= \rho(t)\sin(\phi(t)+\psi(t)) \end{aligned} \tag{1.4}$$

where $\Theta(t)$ is a parameter vector of the signal's amplitude $(\rho(t))$ and phase offset $(\psi(t))$.

In 1967, Viterbi proposed an algorithm for decoding convolutional codes. The algorithm uses forward dynamic programming and is a maximum likelihood decoder [Omura 1969]. This algorithm later became known as the Viterbi Algorithm (VA), and is briefly described in Appendix A.

Sklar [1988] lists the best known generating polynomials for convolutional codes having rate ½, constraint lengths 3 to 9, and rate ⅓, constraint lengths 3 to 8, these codes were determined by Odenwalder in 1970.

§1.3 The Hidden Markov Model

The basic filter theory for HMMs was first presented by Baum and his colleagues in a series of papers in the late 1960s and early 1970s [Baum and Petrie 1966, Baum and Egon 1967, Baum and Sell 1968, Baum *et al.* 1970, Baum 1972]. These papers developed statistical estimation algorithms for discrete Markov processes† observed (hidden) in noise. The model structure became known as a hidden Markov model and since the mid-1980s has become increasingly popular in engineering applications, due in part to introduction and tutorial papers by Rabiner and Juang [1986] and Rabiner [1989].

† The theory of Markov processes originated in the early 1900s [Dynkin, 1965].

Consider the following system known as a HMM (cf. Rabiner [1989]). Assume a finite number, F, of states, $S = \{S_1, S_2, \ldots, S_F\}$, and denote the state at time t by q_t.

The process is assumed to be first order Markovian, that is:

$$\begin{aligned} &\Pr\{q_{t+1} = S_j \mid q_t = S_i, q_{t-1} = S_k, \ldots\} \\ &= \Pr\{q_{t+1} = S_j \mid q_t = S_i\} := a_{ij} \end{aligned} \tag{1.5}$$

where a_{ij}, $1 \le i, j \le F$ is known as the state transition probability from state S_i to state S_j.

The states are observed via a process:

$$U_t = C(q_t) + w_t \tag{1.6}$$

where $C(\cdot)$ is a deterministic function (which in a communications system, would be determined by the modulation type of the signal) and w_t is the noise process. Denote the probability density function of w_t by $f(\cdot)$. The probability of observing U_t given state S_j at time t, $b_j(U_t)$, is given by:

$$\begin{aligned} b_j(U_t) &= \Pr\{U_t \mid q_t = S_j\} \\ &= f(U_t - C(S_j)) \end{aligned} \qquad 1 \le j \le F \tag{1.7}$$

Typically we assume WGN with variance σ^2, thus:

$$b_j(U_t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-1}{2\sigma^2}\left|U_t - C(S_j)\right|^2\right) \qquad 1 \le j \le F \tag{1.8}$$

Finally the model is completed with an initial state distribution vector π, where

$$\pi(i) = \Pr\{q_1 = S_i\} \qquad 1 \le i \le F. \tag{1.9}$$

In signal processing applications, when no *a priori* state information is known, π is usually chosen uniform, i.e. $\pi = (1/F, \ldots, 1/F)$.

The HMM is defined by the parameter set:

$$\lambda = (\pi, \mathrm{A}, \boldsymbol{B}), \tag{1.10}$$

where A and $\boldsymbol{B}$ are the matrices containing the a_{ij}s and $b_j(U_t)$s respectively.

In HMM processing, given an observation sequence and model, a goal may be to estimate a state sequence which is optimal in some sense.

If the objective is to determine, at each separate time, the states which are individually most likely, then the MAP (or minimum variance / conditional mean) state estimates can be determined using the HFBA. The HFBA is briefly described in Appendix B.

If the objective is to determine the most likely state sequence, over all the data, then the ML state sequence estimate can be determined using the VA. The VA is briefly described in Appendix A.

Further details and tutorial introductions to HMMs are provided by Rabiner and Juang [1986], Poritz [1988], and Rabiner [1989].

§1.4 Sensor Array Processing

In Chapters 3 and 4 the concern is with superimposed signals which are transmitted on the same frequency, at the same time, and possibly with the same convolutional encoding scheme, the signals however may be coming from different locations, and with different input information bits. Figure 1.2 shows how an array of sensors may be used to receive such superimposed signals.

In this dissertation the following assumptions are made about the array and signals. The assumptions are: the sensor (antenna) patterns are known; the number of signals incident on the array is known; and the sources are located in the far-field of the array, permitting the narrowband approximation (i.e. planar wavefronts). These assumptions are commonly made in array processing research [Hurt 1990, Haykin 1991] and are generally accepted as valid in most practical situations. This said, research has previously been conducted into: array shape estimation; the reception of wideband signals; and signals which are within finite range, see Haykin [1991] and Hurt [1990] for more details on the research in these areas. Research has also been conducted into determining the number of signals, e.g. Wax [1985] and Wu *et al.* [1995].

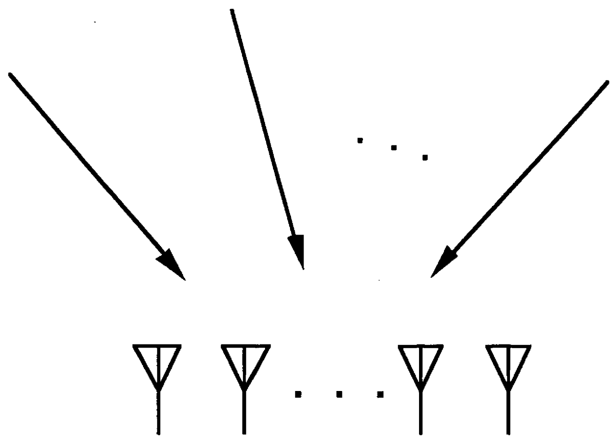

Figure 1.2: Superimposed signals received via an array of sensors.

The signal model for this situation is: Consider L superimposed signals each generated as described above by Eqns. (1.1) to (1.4), except that the input message sequences, $\{b^{(\ell)}(t)\}$, $1 \le \ell \le L$, are generated independently and the generating polynomials $G_m^{(\ell)}(n)$ for each signal may be different. These signals are incident on an array of K sensors. The baseband model of the K-vector array outputs, $\mathbf{U}(t)$, is given by:

$$\mathbf{U}(t) = \mathbf{A}(\Omega)\,\mathbf{Z}(t) + \mathbf{N}(t) \tag{1.11}$$

for $t = 0,\ldots,T-1$, and where $\mathbf{A}(\Omega(t))$ is the so-called direction matrix [Haykin 1991] that depends on the arrival angles $\Omega(t) = \left[\varphi^{(1)}(t),\theta^{(1)}(t),\ldots,\varphi^{(L)}(t),\theta^{(L)}(t)\right]$ of the signals $\mathbf{Z}(t) = \left[z^{(1)}(t), z^{(2)}(t),\ldots,z^{(L)}(t)\right]^T$ (where $\varphi^{(\ell)}(t)$, $\theta^{(\ell)}(t)$ refer to the elevation and azimuth of signal $z^{(\ell)}(t)$ respectively), together with the array geometry. $\mathbf{N}(t)$ is a white Gaussian noise (WGN) process with covariance matrix $\boldsymbol{R}(t)$. The noise is stationary, independent, spatially white and of equal power from sensor to sensor (i.e. $\boldsymbol{R}(t) = \sigma^2\mathbf{I}$).

The processing of superimposed signals generally involves the use of an array of sensors and beamforming techniques. These techniques sequentially estimate the signals parameters and then individually demodulate each signal. The signal models are only introduced in the final demodulation stage. Figure 1.3 shows this sequential procedure.

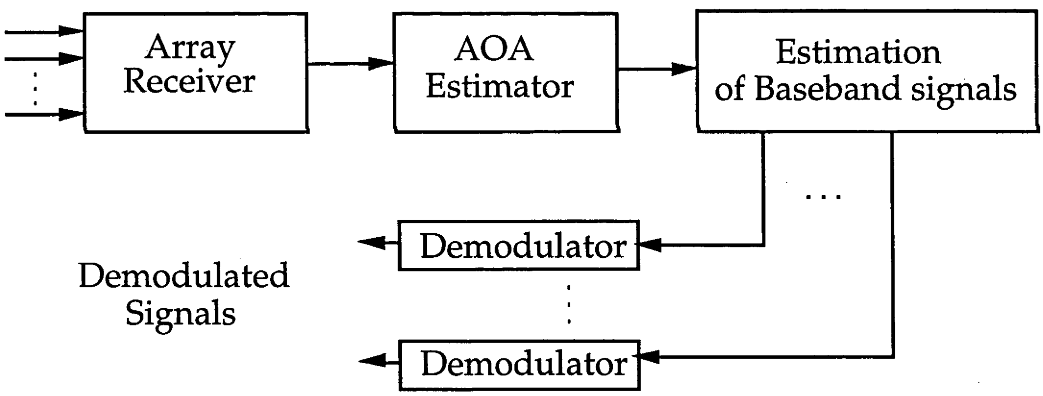

Figure 1.3: Sequential scheme for estimation of superimposed signals.

Most of the parameter estimation algorithms already established have been concerned with direction finding (DF) — the estimation of the AOAs of the signals. Hurt [1990] provides a thorough review of ML estimation techniques in AOA estimation.

Iterative techniques, such as the expectation maximisation (EM) algorithm [Dempster *et al.* 1977] have also been applied to the problem of DF. For example, Feder and Weinstein [1988] introduced the use of the EM algorithm for DF. Miller and Fuhrmann [1990] derived EM algorithms for the ML estimation of the AOAs of multiple narrow-band signals in noise, under both the deterministic and stochastic signal models. Ziskind and Hertz [1993] derived an EM algorithm for Auto-Regressive (AR) processes. Malcolm and White extend Ziskind and Hertz's EM algorithm for general linear Gauss-Markov processes by refining the E-step. Knowledge of the signals characteristics has also been used to improve the AOA estimates, e.g. Trudinger and White [1994] and Talwar *et al.* [1994]. Li and Compton [1993] incorporate knowledge of one or all of the signals waveforms to improve the accuracy of the AOA estimates and Agee *et al.* [1990] use knowledge of the signals' cyclic frequency in order to achieve blind adaptive signal extraction.

In Chapter 3, the sequential method shown in Figure 1.3 is used for performance comparison. The AOA estimates, i.e. $\hat{\Omega}_{\mathrm{ML}}$, are obtained via the method of ML due to Wax [1985], which is briefly described in Appendix E, and the estimates of the baseband signals, i.e. $\hat{\mathbf{Z}}_{\mathrm{ML}}(\mathrm{t})$, are obtained as follows:

$$\hat{\mathbf{Z}}_{\mathrm{ML}}(\mathrm{t}) = \mathbf{A}^{+}\left(\hat{\Omega}_{\mathrm{ML}}\right)\mathbf{A}^{\mathrm{H}}\left(\hat{\Omega}_{\mathrm{ML}}\right)\mathbf{U}(\mathrm{t}) \tag{1.12}$$

where $\mathbf{A}^{+}\left(\hat{\Omega}_{\mathrm{ML}}\right)$ is the pseudo inverse of $\mathbf{A}^{\mathrm{H}}\left(\hat{\Omega}_{\mathrm{ML}}\right)\mathbf{A}\left(\hat{\Omega}_{\mathrm{ML}}\right)$ and is determined via a singular value decomposition, this is in case $\left(\mathbf{A}^{\mathrm{H}}\left(\hat{\Omega}_{\mathrm{ML}}\right)\mathbf{A}\left(\hat{\Omega}_{\mathrm{ML}}\right)\right)^{-1}$ does not exist.

State estimates for each signal, i.e. $\hat{\mathrm{s}}^{(\ell)}(\mathrm{t})$, are then obtained by applying the VA to each ML baseband signal estimate, i.e. $\hat{\mathrm{z}}_{\mathrm{ML}}^{(\ell)}(\mathrm{t})$, individually. This sequential ML method, just outlined, will be referred to as the "ML estimation method" in Chapter 3.

§1.5 Thesis Contributions

The contributions of this thesis and the algorithms (methods) developed are now listed.

- **Estimation of the Structure of a Convolutional Coded Signal:** A technique for determining the constraint length and generating polynomials (structure) of a rate $1/Q$ convolutional coded signal from only the received signal is developed.

- **The use of State Sequence Estimation combined jointly with Parameter Estimation in Array Processing problems:** A method for jointly demodulating superimposed convolutional coded signals and estimating their parameters using knowledge of the models (and structure) of the signals is developed. A modification incorporating the EM algorithm in the parameter estimation step is also developed to reduce some of the computational complexity.

- **A Reduced Complexity Viterbi Algorithm:** Standard reduced state sequence estimators (RSSE), developed for ISI problems, maintain a fixed number of reduced states, however when there are only a few observations and many states (as in the superimposed convolutional coded signals problem), the RSSE techniques may have trouble acquiring lock onto the correct state sequence. The RCVA differs from standard RSSE algorithms in that the number of reduced states is adaptively varied. This modification improves (on average) the probability of acquiring (and maintaining) lock onto the correct state sequence when estimating superimposed convolutional coded signals.

- **Reduced Complexity On-Line Joint Estimator:** An on-line reduced complexity version of the joint state sequence and parameter estimator is developed. This estimator reduces the computational complexity of the above joint estimator by using the RCVA and EM algorithms in on-line modes. It also enables signals with varying AOAs to be successfully tracked.

- **A Viterbi Forward-Backward Algorithm:** The forward backward algorithm is derived from the VA. By presenting the VA's path metric as a forward path probability, a backward path probability is derived in an analogous manner. Combining these two probabilities gives a Viterbi forward-backward algorithm. The VFBA computes an *a posteriori* maximum path probability (AMPP) for each state, at each time, maximised over all state sequences passing through that state, thus yielding uncertainty information about each state estimate, this information is not available directly from the VA. The estimated state sequence obtained via the VFBA is the same as would be obtained via the VA.

- **A Hybrid Viterbi / HMM Forward-Backward Algorithm:** The VFBA above is closely related to the HFBA. This lead to the development of a hybrid algorithm which interpolates between estimating the ML path sequence (via the VA) and estimating MAP state estimates (via the HFBA). The algorithm provides a method for adaptively varying the degree of reliance on path constraints in obtaining state sequence estimates.

§1.6 Thesis Structure

The structure of the thesis is now outlined.

In Chapter 2 a technique for determining the structure (constraint length and generating polynomials) of a convolutional coded signal, from only the received encoded binary data is developed. This technique is confined to the special case of rate $1/Q$ codes and is based on a novel approach recently introduced for blind equalisation of digital communication channels [Moulines *et al.* 1994, Tong *et al.* 1991 and 1993, Xu *et al.* 1994].

In Chapter 3 a method for jointly demodulating and estimating the parameters of superimposed convolutional coded communication signals incident on an antenna array is developed. The method is based on the segmental k-means algorithm (SKMA), a HMM based technique. A brief description of the SKMA is given in Appendix C. The SKMA is an iterative procedure with two steps per iteration. This allows the joint estimation in which the signals are estimated (demodulated), given parameter estimates and the parameter estimates are updated given the demodulated signals. A modification of the parameter estimation step is introduced in order to reduce its computational complexity. The modification applies the EM algorithm for superimposed deterministic signals. Both estimators are then compared to the deterministic ML estimation method described in Section 1.4.

In Chapter 4 a reduced complexity on-line estimator is developed for the problem of jointly demodulating and estimating the parameters of superimposed convolutional coded signals incident on an antenna array. A RCVA is first developed to reduce the computational complexity of jointly demodulating multiple convolutional coded signals via the VA. The RCVA is a modification of the RSSE algorithms developed for ISI problems. The RSSE maintain a fixed number of reduced states, whereas the RCVA adaptively varies the number of reduced states in order to acquire and maintain lock onto the correct state sequences. The on-line estimator jointly uses the RCVA and the EM algorithms in on-line modes. Simulations are used to demonstrate the algorithm and provide comparisons of the computational complexities of the various methods discussed.

In chapter 5 a connection between ML path estimation from noisy observations of a Markovian state sequence and MAP state estimation is developed. The classical VA (ML path estimation) maximises a forward path probability. In an analogous manner a backward path probability is generated which lead to the development of a VFBA. The VFBA computes an *a posteriori* maximum path probability for each state at each time, thus providing a confidence level for each estimated state. Confidence levels are not available from the VA, due to the hard decision backtracking process. The similarity of the VFBA's structure with that of the classical HFBA (used for MAP state estimation) is exploited to obtain a hybrid Viterbi / HMM forward-backward algorithm.

In Chapter 6 a summary of the thesis and conclusions are presented. Suggestions for areas in which further investigation may be conducted and ideas for future research are also outlined.

Chapter 2

Estimating the Structure of a Convolutional Coded Signal

§2.1 Introduction

Convolutional codes (originally proposed by Elias in 1955) are very popular in digital communications (e.g. satellite systems [Heller and Jacobs 1971, Intelsat 1987] and digital mobile phone systems, such as the GSM system [Steele 1992]). This is because convolutional codes provide forward error correction and are optimally decoded by the efficient Viterbi Algorithm [Viterbi 1967]. The VA, like all trellis search algorithms, presumes exact knowledge of the structure of the convolutional encoder (i.e. its constraint length and generating polynomials). In some applications this knowledge may not be available and hence must be extracted from the received signal.

In this chapter a technique is developed for determining the constraint length and the generating polynomials from encoded binary data. The technique is confined to the special case of rate $1/Q$ convolutional codes and is based on a novel approach recently introduced for blind equalisation of digital communication channels [Moulines *et al.* 1994, Tong *et al.* 1991 and 1993, Xu *et al.* 1994].

In Section 2.2 the problem is formulated, Section 2.3 describes the key properties upon which the solution is based, Section 2.4 presents the solution for estimating the structure of the convolutional coded signal and conclusions are presented in Section 2.5.

§2.2 Problem Formulation

Consider a rate $1/Q$, $Q \geq 2$, convolutional encoder with constraint length N, and generating polynomials $\mathbf{g}^{(i)} = \left[g_N^{(i)}, g_{N-1}^{(i)}, \ldots, g_1^{(i)}\right]$, $g_j^{(i)} \in \{0,1\}$, $1 \leq i \leq Q$.

Let $\{b_t\}$, $b_t \in \{0,1\}$, denote the input bits to the convolutional encoder. The encoded bits due to the i^{th} generating polynomial are determined by:

$$u_t^{(i)} = \sum_{j=1}^{N} g_j^{(i)} b_{t-j+1} \qquad i = 1, \ldots, Q \tag{2.1}$$

where the arithmetic is over GF(2), i.e. modulo-2, [Blahut 1984].

The fundamental problem can be now stated as follows. Given T samples of encoded bits $\left\{u_t^{(i)}\right\}_{t=1}^{T}$, $1 \leq i \leq Q$, determine the constraint length N and the generating polynomials $\mathbf{g}^{(i)}$.

To solve the problem we make the following assumptions:

A1: The generating polynomials $g^{(i)}(z) = \sum_{j=0}^{N-1} g_j^{(i)} z^{-j}$ have no common factors.

A2: The constraint length N satisfies N < K, with K some *known* number.

A3: The input bits $\{b_t\}$ are i.i.d. random variables.

A1 implies that the convolutional code is non-catastrophic, which is generally the case in practice. *A2* implies that the upper limit on the constraint length is known. *A3* is very common and frequently used in digital communications.

§2.3 The Key Properties

Using a sliding window of length K, Eqn. (2.1) can be rewritten in the following matrix form (where $\otimes$ stands for matrix multiplication with the arithmetic over GF(2)):

$$\begin{bmatrix} u_1^{(i)} & u_2^{(i)} & \cdots & u_{T-K+1}^{(i)} \\ u_2^{(i)} & u_3^{(i)} & \cdots & u_{T-K+2}^{(i)} \\ \vdots & \vdots & \vdots & \vdots \\ u_K^{(i)} & u_{K+1}^{(i)} & \cdots & u_T^{(i)} \end{bmatrix} = \begin{bmatrix} g_N^{(i)} & g_{N-1}^{(i)} & \cdots & g_1^{(i)} & & & 0 \\ & g_N^{(i)} & g_{N-1}^{(i)} & \cdots & g_1^{(i)} & & \\ & & \ddots & \ddots & \cdots & \ddots & \\ 0 & & & g_N^{(i)} & g_{N-1}^{(i)} & \cdots & g_1^{(i)} \end{bmatrix}$$

$$\otimes \begin{bmatrix} b_1 & b_2 & \cdots & b_{T-(N+K-2)} \\ b_2 & b_3 & \cdots & b_{T-(N+K-3)} \\ \vdots & \vdots & \vdots & \vdots \\ b_{N+K-1} & b_{N+K-2} & \cdots & b_T \end{bmatrix} \tag{2.2}$$

Denoting by $\mathbf{G}^{(i)}$ the Kx(N+K-1) Toeplitz matrix of the i^{th} generating polynomial as:

$$\mathbf{G}^{(i)} = \begin{bmatrix} g_N^{(i)} & g_{N-1}^{(i)} & \cdots & g_1^{(i)} & 0 & & \cdots & 0 \\ 0 & g_N^{(i)} & g_{N-1}^{(i)} & \cdots & g_1^{(i)} & 0 & \cdots & \\ \vdots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \vdots \\ & & & & & & & 0 \\ 0 & \cdots & & 0 & g_N^{(i)} & g_{N-1}^{(i)} & \cdots & g_1^{(i)} \end{bmatrix} \tag{2.3}$$

and by $\mathbf{U}^{(i)}$ and $\mathbf{B}$ the Kx(T-K+1) and (N+K-1)x(T-K+1) Hankel matrices as:

$$\mathbf{U}^{(i)} = \begin{bmatrix} u_1^{(i)} & u_2^{(i)} & \cdots & u_{T-K+1}^{(i)} \\ u_2^{(i)} & u_3^{(i)} & \cdots & u_{T-K+2}^{(i)} \\ \vdots & \vdots & \vdots & \vdots \\ u_K^{(i)} & u_{K+1}^{(i)} & \cdots & u_T^{(i)} \end{bmatrix} \tag{2.4}$$

and

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & \cdots & b_{T-(N+K-2)} \\ b_2 & b_3 & \cdots & b_{T-(N+K-3)} \\ \vdots & \vdots & \vdots & \vdots \\ b_{N+K-1} & b_{N+K-2} & \cdots & b_T \end{bmatrix} \tag{2.5}$$

Eqn. (2.2) can be rewritten as:

$$\mathbf{U}^{(i)} = \mathbf{G}^{(i)} \otimes \mathbf{B} \qquad i = 1 , \ldots , Q \tag{2.6}$$

Concatenating the matrices $\mathbf{U}^{(i)}$, i = 1 , ... , Q, gives:

$$\mathbf{U} = \mathbf{G} \otimes \mathbf{B} \tag{2.7}$$

where **U** is the KQx(T-N+1) matrix:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}^{(1)} \\ \vdots \\ \mathbf{U}^{(Q)} \end{bmatrix} \tag{2.8}$$

and **G** is the KQx(N+K-1) matrix:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}^{(1)} \\ \vdots \\ \mathbf{G}^{(Q)} \end{bmatrix} \tag{2.9}$$

The two key properties upon which the solution is based are now stated.

Theorem 2.1: Under assumptions *A1* and *A2*, the matrix **G** is full column rank.

Proof: See Tong *et al.* [1991 and 1993]. □

Theorem 2.2: Under assumptions *A1* - *A3*, the rank of $\mathbf{U}$ is:

$$Rank_{GF(2)}\,\mathbf{U} = N+K-1 \tag{2.10}$$

with probability which approaches one as $T \rightarrow \infty$.

Proof: From Theorem 2.1:

$$Rank_{GF(2)}\,\mathbf{G} = N+K-1 \tag{2.11}$$

Now from *A3* it follows that if $T \rightarrow \infty$ the matrix $\mathbf{B}$ is full-row rank with probability one, i.e.

$$Rank_{GF(2)}\,\mathbf{B} = N+K-1 \tag{2.12}$$

The result now follows readily by using (2.11) and (2.12) in (2.7).

Note that for Theorem 2.2 to hold with probability close to one it suffices that the number of columns in $\mathbf{B}$ is much larger than the number of rows, that is T-K+1 >> N+K-1, which implies T >> 3K+1. □

Theorem 2.3: Under assumptions *A1* - *A2*, the left null-space of $\mathbf{G}$ determines the generating polynomials coefficients $\{\mathbf{g}^{(i)}\}$ up to a multiplicative constant.

Proof: See Moulines *et al.* [1994]. □

§2.4 The Solution

The constraint length N is readily determined from the rank of $\mathbf{U}$.

To determine the generating polynomials, let $\mathbf{w}_j$ be a left null-vector of $\mathbf{U}$, i.e.

$$\mathbf{w}_j \otimes \mathbf{U} = 0 \qquad j = 1, \ldots, KQ - (N+K-1) \tag{2.13}$$

From Eqn. (2.7):

$$\mathbf{w}_j \otimes \mathbf{G} \otimes \mathbf{B} = 0 \tag{2.14}$$

which implies, since **B** is full-row rank, that:

$$\mathbf{w}_j \otimes \mathbf{G} = 0 \tag{2.15}$$

Let $\mathbf{w}_j^{(i)}$, $1 \leq i \leq Q$, be the i^{th} 1xK block of $\mathbf{w}_j$, i.e.

$$\mathbf{w}_j = \left[\mathbf{w}_j^{(1)}, \mathbf{w}_j^{(2)}, \ldots, \mathbf{w}_j^{(Q)}\right] \tag{2.16}$$

where

$$\mathbf{w}_j^{(i)} = \left[w_{j1}^{(i)}, w_{j2}^{(i)}, \ldots, w_{jK}^{(i)}\right] \tag{2.17}$$

Now (as can be readily verified) the following structural relation holds,

$$\mathbf{w}_j \otimes \mathbf{G} = \mathbf{g} \otimes \mathbf{W}_j \qquad j = 1, \ldots, KQ - (N+K-1) \tag{2.18}$$

where

$$\mathbf{g} = \left[\mathbf{g}^{(1)}, \mathbf{g}^{(2)}, \ldots, \mathbf{g}^{(Q)}\right] \tag{2.19}$$

with

$$\mathbf{g}^{(i)} = \left[g_N^{(i)}, g_{N-1}^{(i)}, \ldots, g_1^{(i)}\right] \tag{2.20}$$

and

$$\mathbf{W}_j = \begin{bmatrix} \mathbf{W}_j^{(1)} \\ \vdots \\ \mathbf{W}_j^{(Q)} \end{bmatrix} \tag{2.21}$$

with $\mathbf{W}_j^{(i)}$ being the Nx(N+K-1) Toeplitz matrix:

$$\mathbf{W}_j^{(i)} = \begin{bmatrix} w_{j1}^{(i)} & \cdots & w_{jK}^{(i)} & 0 & \cdots & 0 \\ 0 & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & w_{j1}^{(i)} & \cdots & w_{jK}^{(i)} \end{bmatrix} \tag{2.22}$$

From Eqns. (2.15) and (2.18),

$$\mathbf{g} \otimes \mathbf{W}_j = 0 \qquad j = 1, \ldots, KQ - (N+K-1) \tag{2.23}$$

which can be rewritten as

$$\mathbf{g} \otimes \mathbf{W} = 0 \tag{2.24}$$

where

$$\mathbf{W} = \left[\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{KQ-(N+K-1)}\right] \tag{2.25}$$

By Theorem 2.3, $\mathbf{g}$ is the single left null-vector of $\mathbf{W}$ and the $\mathbf{g}^{(i)}$s obtained from $\mathbf{g}$, Eqns. (2.19) and (2.20), are the generating polynomials.

The solution can be summarised as follows:

(1) Determine the rank of $\mathbf{U}$.

(2) Determine N using Eqn. (2.10).

(3) Compute a left null-vectors $\mathbf{w}_j$ of $\mathbf{U}$, $j = 1, ..., KQ - (N+K-1)$.

(4) Using $\mathbf{w}_j$ construct $\mathbf{W}_j^{(i)}$, $i = 1, ..., Q$ using Eqn. (2.22).

(5) Construct $\mathbf{W}_j$ using Eqn. (2.21).

(6) Construct $\mathbf{W}$ using Eqn. (2.25).

(7) Compute $\mathbf{g}$ (and hence the $\mathbf{g}^{(i)}$ s) as the left null-vector of $\mathbf{W}$.

Notice that steps (1), (3) and (7) are over GF(2).

§2.5 Conclusion

The method described in this chapter does not require the received encoded data to be noiseless, rather it requires that hard decisions on a set of T consecutive samples of the noisy data be error-free. If any errors occur in the set of T consecutive samples due to the hard decisions on the noisy data — the method fails.

Chapter 3

Joint Demodulation and Parameter Estimation

§3.1 Introduction

The problem addressed in this chapter and in Chapter 4 is the spatial filtering of superimposed convolutional coded QPSK communications signals incident on a Uniform Linear Array (ULA) of sensors. A method for jointly demodulating the signals and estimating the signals' parameters, e.g. amplitude, phase offset and angle of arrival (AOA), is developed.

The estimation of superimposed signals incident on an array of sensors is not new, see Chapter 1, Section 1.4 for details. This chapter investigates the estimation of superimposed signals modelled as Markovian sequences. These Markovian sequences (convolutional coded signals) have strongly constrained path sequences and therefore the estimation procedure should yield valid path constrained sequences [Sklar 1988]. Knowledge of the signals' models is assumed (i.e. each signal is convolutional coded with known constraint length and generating polynomials).

The segmental k-means algorithm (SKMA), a HMM based technique, is applied to the problem (see Appendix C for a brief description of the SKMA). This algorithm is an iterative procedure requiring two steps per iteration. The first step (segmentation step) involves computing the ML state sequence for all the signals (demodulating the signals) given estimates of the signals' parameters, this is accomplished using dynamic programming via the VA. The second step (optimisation step) uses the estimated ML state sequences to refine estimates of the signals' parameters by maximising the state-optimised log likelihood function with respect to the parameters.

A modification of the method which decreases the computational complexity of this problem and hence the processing time is also described. The modification makes use of the EM algorithm for superimposed deterministic signals (briefly described in Appendix D) in the optimisation step of the SKMA.

The methods developed are shown to improve the demodulation of the signals and in addition improves the accuracy in estimating the AOAs, when compared to the deterministic ML estimation method outlined in Chapter 1, Section 1.4. Superior resolution capability for signals that are closely spaced is also demonstrated. The improvements are shown to be significant for the examples discussed but are achieved through an increase in the computational complexity of the problem. Monte Carlo simulations are used to demonstrate the results and improvements obtained.

This chapter is organised as follows: In Section 3.2 a single signal (not incident on an array) is used to provide a simple example for explaining and testing the SKMA. The algorithm is used to jointly demodulate the signal and estimate the signal's amplitude and phase offset. In Section 3.3, the SKMA is described for a single signal incident on an array (Section 3.3.1) and for superimposed signals incident on an array (Section 3.3.2). The expressions for demodulating the signal(s) and estimating the signal(s)'s parameters are derived. The EM modification to the optimisation (parameter estimation) step is also described in Section 3.3.2. Simulation studies are presented in Section 3.4 and conclusions drawn from these are given in Section 3.5.

§3.2 Single Signal

The observed signal model for a single convolutional coded QPSK signal (defined in Chapter 1, Section 1.2) is given by:

$$z(s(t),\Theta(t)) = x(t) + i\,y(t) \qquad \text{with} \qquad \begin{aligned} x(t) &= \rho(t)\cos(\phi(t)+\psi(t)) + w(t) \\ y(t) &= \rho(t)\sin(\phi(t)+\psi(t)) + v(t) \end{aligned} \tag{3.1}$$

where $w(t), v(t)$ are i.i.d. WGN processes with zero mean and variance σ^2.

The problem of interest is to jointly estimate the parameters ($\rho(t)$ and $\psi(t)$ which can be slowly varying with respect to time, but are considered constant for some block length T†) and the message sequence $\{\hat{b}(t)\}$ (see Chapter 1, Eqn. 1.2) from the observed data sequence $\{z(t)\} = \{z(s(t),\Theta)\}$.

The SKMA is a parameter estimation algorithm which involves data sequence modelling. This algorithm was examined as a means of jointly demodulating the signal and estimating the signal's parameters. Juang and Rabiner [1990] provide sufficient conditions for global convergence of the SKMA, however these conditions are difficult to test and impose on the problems considered in this thesis. A local convergence property for the algorithm applied to the problems in this thesis was conjectured from the empirical evidence obtained, however a localised stability analysis is intractable.

In the segmentation step of the algorithm, the objective (in a fixed interval demodulation problem) is to estimate the i^th^ iterative ML state sequence $\{\hat{s}_i(t)\}$:

$$\{\hat{s}_i(t)\} = \hat{s}_i(0),\ldots,\hat{s}_i(T-1) = \underset{s(0),\ldots,s(T-1)}{\arg\max}\ \Pr\{s(0),\ldots,s(T-1)\,|\,z(0),\ldots,z(T-1);\hat{\rho}_i,\hat{\psi}_i\} \tag{3.2}$$

† The time index of the parameters are removed for the rest of this chapter.

for some signal block length T and i^{th} estimates $\hat{\rho}_i$ and $\hat{\psi}_i$. It has been shown that this problem can be solved using dynamic programming, resulting in the VA. The VA is used in the segmentation step of the SKMA in preference to alternative estimation algorithms (Bahl *et al.* [1974] and Baum *et al.* [1970]) for similar reasons as those stated in Juang and Rabiner [1990], i.e. the requirement to reduce the computational complexity, etc., of the problem.

The i^{th} estimated ML sequence $\hat{s}_i(t)=[\hat{c}_i(t-N+1),\ldots,\hat{c}_i(t-1),\hat{c}_i(t)]$, $0\le t<T$, where $\hat{c}_i(m)=0$, $m<0$, is used to obtain an estimate of the message sequence $\{\hat{b}_i(t)\}$. If the demodulated message bit ($\hat{b}_i(t)$) is obtained from the MSB of $\hat{s}_i(t+N-1)$, instead of the LSB of $\hat{s}_i(t)$, a smoothed sequence can be obtained as shown below:

$$\hat{b}_i(t)=\hat{c}_i(t+N-1), \qquad 0\le t\le T-N. \tag{3.3}$$

The demodulated message sequence obtained from Eqn. (3.3) may be improved if more accurate estimates of the signal's amplitude and phase offset can be obtained. The optimisation step of the algorithm uses the i^{th} estimated ML state sequence $\{\hat{s}_i(t)\}$ and observation sequence $\{z(t)\}$ to generate the next parameter estimates $(\hat{\rho}_{i+1},\hat{\psi}_{i+1})$. These estimates maximise the state-optimised log likelihood function:

$$\left(\hat{\rho}_{i+1},\hat{\psi}_{i+1}\right)=\arg\max_{\rho,\psi}\left\{\log \Pr\left\{\{z(t)\}\middle|\{\hat{s}_i(t)\};\rho,\psi\right\}\right\}. \tag{3.4}$$

For the convolutional coded QPSK signal, Eqn. (3.4) is written as:

$$\begin{aligned}\left(\hat{\rho}_{i+1},\hat{\psi}_{i+1}\right)&=\arg\max_{\rho,\psi}\left\{-T\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}\sum_{t=0}^{T-1}\left|z(t)-\rho e^{i\left(\hat{\phi}_i(t)+\psi\right)}\right|^2\right\}\\ &=\arg\max_{\rho,\psi}\left\{-T\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}\sum_{t=0}^{T-1}\left[\left(x(t)-\rho\cos\left(\hat{\phi}_i(t)+\psi\right)\right)^2\right.\right.\\ &\qquad\left.\left.+\left(y(t)-\rho\sin\left(\hat{\phi}_i(t)+\psi\right)\right)^2\right]\right\}\end{aligned} \tag{3.5}$$

$$\text{where} \quad \hat{\phi}_i(t) = \frac{2\pi}{M}\sum_{m=0}^{M-1} 2^m \sum_{n=0}^{N} \hat{b}_i(t-n)G_m(n), \text{ and}$$

$\hat{b}_i(t)$ is obtained from $\hat{s}_i(t)$.

The updated estimates of the signal's parameters are determined from Eqn. (3.5) using basic calculus and are given in closed form by:

$$\hat{\rho}_{i+1} = \frac{1}{T}\sqrt{\left[(Y_{im}+X_{re})^2 + (Y_{re}-X_{im})^2\right]}$$

$$\hat{\psi}_{i+1} = \arctan\left(\frac{Y_{re}-X_{im}}{Y_{im}+X_{re}}\right)$$

$$\text{where} \quad Y_{re} = \sum_{t=0}^{T-1} y(t)\cos\left(\hat{\phi}_i(t)\right),\ Y_{im} = \sum_{t=0}^{T-1} y(t)\sin\left(\hat{\phi}_i(t)\right), \tag{3.6}$$

$$X_{re} = \sum_{t=0}^{T-1} x(t)\cos\left(\hat{\phi}_i(t)\right),\ X_{im} = \sum_{t=0}^{T-1} x(t)\sin\left(\hat{\phi}_i(t)\right).$$

These updated estimates ($\hat{\rho}_{i+1}$ and $\hat{\psi}_{i+1}$) are used in the next segmentation step. The process is repeated until an imposed convergence criterion is satisfied, for example when the difference of two consecutive estimates of both $\hat{\rho}$ and $\hat{\psi}$ are smaller than or equal to given thresholds (ξ), i.e.

$$\left|\hat{\rho}_{i+1} - \hat{\rho}_i\right| \le \xi_\rho \text{ and } \left|\hat{\psi}_{i+1} - \hat{\psi}_i\right| \le \xi_\psi. \tag{3.7}$$

§3.3 Array Processing

§3.3.1 Single Signal

The observed signal model for a single signal incident on an array of sensors is given in Chapter 1, Section 1.4 with L = 1.

The problem of interest here is to jointly estimate the amplitude (ρ), phase offset (ψ), arrival angles $(\varphi,\theta)^{\ddagger}$ and the message sequence $\{b(t)\}$ from the observed data sequence $\{\mathbf{U}(t)\}$.

Again, the SKMA method is used for demodulating the signal and estimating the amplitude, phase offset and AOA of the signal. The approach is similar to the method described above in Section 3.2 with minor changes to Eqn. (3.2) and Eqns. (3.4) to (3.6). These new equations are as follows:

The i^{th} estimated ML state sequence given in Eqn. (3.2) is now estimated by determining:

$$\{\hat{s}_i(t)\} = \hat{s}_i(0), \ldots, \hat{s}_i(T-1) = \underset{s(0),\ldots,s(T-1)}{\arg\max} \Pr\left\{s(0), \ldots, s(T-1) \middle| \mathbf{U}(0), \ldots, \mathbf{U}(T-1); \hat{\rho}_i, \hat{\psi}_i, \hat{\Omega}_i\right\} \tag{3.8}$$

for some block length T and i^{th} estimates of $\hat{\rho}_i$, $\hat{\psi}_i$ and $\hat{\Omega}_i$. The VA provides a solution to the problem.

An improved estimate of the ML state sequence can be obtained if more accurate estimates of the signal's amplitude, phase offset and AOA are obtained. Thus, as in Section 3.2, the state-optimised log likelihood function is used and Eqn. (3.4) is rewritten as:

$$\left(\hat{\rho}_{i+1}, \hat{\psi}_{i+1}, \hat{\Omega}_{i+1}\right) = \underset{\rho,\psi,\Omega}{\arg\max}\left\{\log \Pr\left\{ \{\mathbf{U}(t)\} \middle| \{\hat{s}_i(t)\}; \rho, \psi, \Omega \right\}\right\}. \tag{3.9}$$

The output vector of the array, $\mathbf{U}(t)$, is written in quadrature baseband form as $u(t,k) = x(t,k) + i\, y(t,k)$ for each sensor, k. For a convolutional coded QPSK plane wave signal, Eqn. (3.9) is rewritten as follows (using the azimuth angle $\theta^{\dagger}$):

---

‡ Again these parameters are assumed to be constant for some block length T, and thus the time index is not shown.

† The following mathematical descriptions could be extended to also estimate the elevation angle, φ.

$$\left(\hat{\rho}_{i+1},\hat{\psi}_{i+1},\hat{\theta}_{i+1}\right)=\arg\max_{\rho,\psi,\theta}\left\{-T\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\left[\left(x(t,k)-\rho\cos\left(\hat{\phi}_i(t)+\psi+k\omega\right)\right)^2\right.\right.$$

$$\left.\left.+\left(y(t,k)-\rho\sin\left(\hat{\phi}_i(t)+\psi+k\omega\right)\right)^2\right]\right\} \qquad (3.10)$$

where $\hat{\phi}_i(t)=\frac{2\pi}{M}\sum_{m=0}^{M-1}2^m\sum_{n=0}^{N}\hat{b}_i(t-n)G_m(n)$

$\hat{b}_i(t)$ is obtained from $\hat{s}_i(t)$

$\omega=\frac{2\pi d}{\lambda}\cos(\theta)$

$d=$ spacing between sensors

$\lambda=$ signal carrier wavelength.

An updated estimate of the parameters can now be solved for as follows:

$$\hat{\theta}_{i+1}=\max_{\theta}\left\{\frac{-1}{2\sigma^2}\left[XY+\left(\frac{1-2TK}{(TK)^2}\right)\left[\left(Y_{im}+X_{re}\right)^2+\left(Y_{re}-X_{im}\right)^2\right]\right]\right\}$$

where
$$Y_{re}=\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}y(t,k)\cos\left(\hat{\phi}_i(t)+k\omega\right),$$

$$Y_{im}=\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}y(t,k)\sin\left(\hat{\phi}_i(t)+k\omega\right), \qquad (3.11)$$

$$X_{re}=\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}x(t,k)\cos\left(\hat{\phi}_i(t)+k\omega\right),$$

$$X_{im}=\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}x(t,k)\sin\left(\hat{\phi}_i(t)+k\omega\right),$$

$$XY=\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\left(x(t,k)^2+y(t,k)^2\right).$$

$$\hat{\rho}_{i+1}=\frac{1}{TK}\sqrt{\left[\left(Y_{im}+X_{re}\right)^2+\left(Y_{re}-X_{im}\right)^2\right]}\Bigg|_{\hat{\theta}_{i+1}},$$

$$\hat{\psi}_{i+1}=\arctan\left(\frac{Y_{re}-X_{im}}{Y_{im}+X_{re}}\right)\Bigg|_{\hat{\theta}_{i+1}}. \qquad (3.12)$$

The above maximisation can not be written in closed form and therefore $\hat{\theta}_{i+1}$ is approximated using numerical techniques [Burden and Faires 1985] and/or a grid search within a given search region of H intervals (a grid search is used throughout this thesis). $\hat{\rho}_{i+1}$ and $\hat{\psi}_{i+1}$ can be evaluated in closed form once $\hat{\theta}_{i+1}$ has been evaluated.

As in Section 3.2, the algorithm is repeated until the imposed convergence criterion, Eqn. (3.7), is satisfied as well as:

$$\left|\hat{\theta}_{i+1} - \hat{\theta}_i\right| \leq \xi_\theta. \tag{3.13}$$

§3.3.2 Superimposed Signals

The observed signal model for L superimposed signals is given by Eqn. (1.11). For optimal demodulation, these L signals can be considered to be equivalent to a first order Markov process with F^L states given by:

$$\begin{aligned}
S(t) &= \left[s^{(1)}(t), s^{(2)}(t), \ldots, s^{(L)}(t)\right] \\
&= \left[b^{(1)}(t-N+1), \ldots, b^{(1)}(t-1), b^{(1)}(t), b^{(2)}(t-N+1), \ldots,\right. \\
&\qquad \left. b^{(2)}(t-1), b^{(2)}(t), \ldots, b^{(L)}(t-N+1), \ldots, b^{(L)}(t-1), b^{(L)}(t)\right],
\end{aligned} \tag{3.14}$$

with transition probabilities:

$$\begin{aligned}
&\Pr\left\{S(t) = \left[c^{(1)}(N-1), \ldots, c^{(1)}(0), c^{(2)}(N-1), \ldots, c^{(2)}(0), \ldots, c^{(L)}(N-1), \ldots, c^{(L)}(0)\right]\right. \\
&\quad \left| S(t-1) = \left[d^{(1)}(N-1), \ldots, d^{(1)}(0), d^{(2)}(N-1), \ldots, d^{(2)}(0), \ldots, d^{(L)}(N-1), \ldots,\right.\right. \\
&\quad \left.\left. d^{(L)}(0)\right]\right\} = \begin{cases} a_{d^{(\ell)}(0)c^{(\ell)}(0)} & c^{(\ell)}(j) = d^{(\ell)}(j-1);\ 1 \leq j \leq N-1;\ 1 \leq \ell \leq L \\ 0 & \text{else} \end{cases}
\end{aligned} \tag{3.15}$$

where $a_{d^{(\ell)}(0)c^{(\ell)}(0)} = 0.5^L$ when the L input message sequences $\{b^{(\ell)}(t)\}$ are binary, i.i.d. & mutually independent, and the c(j)s and d(j)s are again dummy variables as in Eqn. (1.3).

The SKMA is used to demodulate the signals and estimate their parameters. The segmentation step estimates the i^{th} iterative ML state sequence of the signals,

$$\begin{aligned}\left\{\hat{S}_i(t)\right\} &= \hat{S}_i(0), \ldots, \hat{S}_i(T-1) \\ &= \underset{S(0),\ldots,S(T-1)}{\arg\max} \Pr\left\{S(0), \ldots, S(T-1) \,\middle|\, \mathbf{U}(0), \ldots, \mathbf{U}(T-1); \hat{P}_i, \hat{\Psi}_i, \hat{\Omega}_i\right\}\end{aligned} \tag{3.16}$$

for some block length T and i^{th} estimates $\hat{P}_i = \left\{\hat{\rho}_i^{(1)}, \hat{\rho}_i^{(2)}, \ldots, \hat{\rho}_i^{(L)}\right\}$, $\hat{\Psi}_i = \left\{\hat{\psi}_i^{(1)}, \hat{\psi}_i^{(2)}, \ldots, \hat{\psi}_i^{(L)}\right\}$ and $\hat{\Omega}_i = \left\{\hat{\varphi}_i^{(1)}, \hat{\theta}_i^{(1)}, \hat{\varphi}_i^{(2)}, \hat{\theta}_i^{(2)}, \ldots, \hat{\varphi}_i^{(L)}, \hat{\theta}_i^{(L)}\right\}$.

The sequence $\left\{\hat{S}_i(t)\right\}$ is estimated as one signal that has F^L states, using the VA. From Eqn. (3.14), the L estimated ML state sequences $\left\{\hat{s}_i^{(\ell)}(t)\right\}$ and hence the L demodulated message sequences $\left\{\hat{b}_i^{(\ell)}(t)\right\}$, $1 \le \ell \le L$ are then obtained.

To improve the demodulation of the signals, updated estimates of the signals' parameters (i.e. $\hat{P}_{i+1}, \hat{\Psi}_{i+1}$ and $\hat{\Omega}_{i+1}$) are determined to maximise the state-optimised log likelihood function:

$$\left(\hat{P}_{i+1}, \hat{\Psi}_{i+1}, \hat{\Omega}_{i+1}\right) = \underset{P,\Psi,\Omega}{\arg\max}\left\{\log \Pr\left\{\{\mathbf{U}(t)\} \,\middle|\, \left\{\hat{S}_i(t)\right\}; P, \Psi, \Omega\right\}\right\}. \tag{3.17}$$

For the convolutional coded QPSK plane wave signals, Eqn. (3.17) is written as follows, with $\hat{\Omega}$ again replaced by the signals' azimuth angles $\hat{\Theta} = \left\{\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \ldots, \hat{\theta}^{(L)}\right\}$:

$$\begin{aligned}\left(\hat{P}_{i+1}, \hat{\Psi}_{i+1}, \hat{\Theta}_{i+1}\right) &= \underset{P,\Psi,\Theta}{\arg\max}\left\{-T\log\left(\sigma\sqrt{2\pi}\right) - \frac{1}{2\sigma^2}\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\left|u(t,k) - \sum_{\ell=1}^{L}\rho^{(\ell)} e^{i\left(\hat{\phi}_i^{(\ell)}(t)+\psi^{(\ell)}+k\omega^{(\ell)}\right)}\right|^2\right\} \\ &= \underset{P,\Psi,\Theta}{\arg\max}\left\{-T\log\left(\sigma\sqrt{2\pi}\right) - \frac{1}{2\sigma^2}\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\left[\left(x(t,k) - \sum_{\ell=1}^{L}\rho^{(\ell)}\cos\left(\hat{\phi}_i^{(\ell)}(t) + \psi^{(\ell)} + k\omega^{(\ell)}\right)\right)^2\right.\right. \\ &\qquad \left.\left. + \left(y(t,k) - \sum_{\ell=1}^{L}\rho^{(\ell)}\sin\left(\hat{\phi}_i^{(\ell)}(t) + \psi^{(\ell)} + k\omega^{(\ell)}\right)\right)^2\right]\right\}\end{aligned} \tag{3.18}$$

$$\text{where } \hat{\phi}_i^{(\ell)}(t) = \frac{2\pi}{M}\sum_{m=0}^{M-1} 2^m \sum_{n=0}^{N} \hat{b}_i^{(\ell)}(t-n)G_m^{(\ell)}(n)$$

and $\hat{b}_i^{(\ell)}(t)$ is obtained from $\hat{s}_i^{(\ell)}(t)$

$$\omega^{(\ell)} = \frac{2\pi d}{\lambda}\cos\left(\theta^{(\ell)}\right)$$

d = spacing between sensors

λ = signal carrier wavelength.

There is no closed form solution to Eqn. (3.18), however a number of numerical techniques [Burden and Faires 1985] can be used to determine the values of $\hat{P}_{i+1}, \hat{\Psi}_{i+1}$ and $\hat{\Theta}_{i+1}$ which maximise Eqn. (3.18). An L-dimensional grid search (with finite intervals) over the range of possible values could also be applied to approximately maximise Eqn. (3.18). Both of these methods are computationally very intensive.

The direct maximisation in Eqn. (3.18) is a non linear optimisation problem in many unknowns. The EM approach proposed by Feder and Weinstein [1988] (Appendix D) is used to reduce the complexity of this problem by replacing a difficult maximisation problem with an iterative sequence of simpler maximisation problems. This EM method is detailed below. The EM method iteratively increases the likelihood in order to approximate the maximisation of Eqn. (3.18). As stated above Eqn. (3.18) has no closed form solution, thus the exact maximisation is not obtainable. That said, the approximate maximisation is sufficient as the VA is robust to small errors in the parameter estimates, this is evident from the simulations in Section 3.4.

The parameter values used in the segmentation step of the SKMA are used to initialise the parameter values for the EM algorithm, i.e.

Initialisation:

$$\left(\hat{P}_{i,0}, \hat{\Psi}_{i,0}, \hat{\Theta}_{i,0}\right) = \left(\hat{P}_i, \hat{\Psi}_i, \hat{\Theta}_i\right) \tag{3.19}$$

The EM algorithm updates the signals' parameters by iterating between:

a) The Expectation step (E-step)

Estimate an observation sequence for each signal $\{\hat{\mathbf{u}}_{i,b}^{(\ell)}(t)\}$, $1 \le \ell \le L$, from the observed data sequence $\{\mathbf{U}(t)\}$ given the parameter estimates $\hat{\mathrm{P}}_{i,b}, \hat{\Psi}_{i,b}$ and $\hat{\Theta}_{i,b}$ and state sequence estimates $\{\hat{S}_i(t)\}$.

b) The Maximisation step (M-step)

Update estimates of the signals' parameters ($\hat{\mathrm{P}}_{i,b+1}, \hat{\Psi}_{i,b+1}$ and $\hat{\Theta}_{i,b+1}$) by maximising the log-likelihood functions for each signal's observation sequence $\{\hat{\mathbf{u}}_{i,b}^{(\ell)}(t)\}$.

These steps are described below:

Recursion:

For each signal ℓ, $1 \le \ell \le L$

E-step: For each sensor k, $0 \le k \le K-1$

$$\hat{u}_{i,b}^{(\ell)}(t,k) = \hat{\rho}_{i,b}^{(\ell)} e^{i\left(\hat{\phi}_i^{(\ell)}(t)+\psi_{i,b}^{(\ell)}+k\omega_{i,b}^{(\ell)}\right)} + \zeta^{(\ell)}\left[U(t,k) - \sum_{r=1}^{L} \hat{\rho}_{i,b}^{(r)} e^{i\left(\hat{\phi}_i^{(r)}(t)+\hat{\psi}_{i,b}^{(r)}+k\hat{\omega}_{i,b}^{(r)}\right)}\right] \tag{3.20}$$

M-step:

$$\left(\hat{\rho}_{i,b+1}^{(\ell)}, \hat{\psi}_{i,b+1}^{(\ell)}, \hat{\theta}_{i,b+1}^{(\ell)}\right) = \arg\max_{\rho,\psi,\theta}\left\{\frac{-1}{2\sigma^2}\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\left|\hat{u}_{i,b}^{(\ell)}(t,k) - \rho e^{i\left(\hat{\phi}_i^{(\ell)}(t)+\psi+k\omega\right)}\right|^2\right\} \tag{3.21}$$

$$\begin{aligned} &= \arg\max_{\rho,\psi,\theta}\left\{\frac{-1}{2\sigma^2}\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\left[\left(\hat{x}_{i,b}^{(\ell)}(t,k) - \rho\cos\left(\hat{\phi}_i^{(\ell)}(t)+\psi+k\omega\right)\right)^2\right.\right. \\ &\qquad \left.\left. + \left(\hat{y}_{i,b}^{(\ell)}(t,k) - \rho\sin\left(\hat{\phi}_i^{(\ell)}(t)+\psi+k\omega\right)\right)^2\right]\right\}. \end{aligned} \tag{3.22}$$

where

$$\zeta^{(\ell)} \geq 0 \text{ with } \sum_{\ell=1}^{L} \zeta^{(\ell)} = 1, \text{ and}$$

$\hat{\phi}_i^{(\ell)}(t)$ and ω are similarly defined as in Eqn. (3.18) above.

Eqn. (3.22) is now in an expanded form similar to Eqn. (3.18) except the terms summed over ℓ are replaced by single terms. From this the following equations can be derived for the next estimate of the parameters:

$$\hat{\theta}_{i,b+1}^{(\ell)} = \max_{\theta}\left\{\frac{-1}{2\sigma^2}\left[XY + \left(\frac{1-2TK}{(TK)^2}\right)\left[\left(Y_{im}^{i,b} + X_{re}^{i,b}\right)^2 + \left(Y_{re}^{i,b} - X_{im}^{i,b}\right)^2\right]\right]\right\} \tag{3.23}$$

$$\text{where} \quad Y_{re}^{i,b} = \sum_{t=0}^{T-1}\sum_{k=0}^{K-1} y_{i,b}^{(\ell)}(t,k)\cos\left(\hat{\phi}_i^{(\ell)}(t) + k\omega\right),$$

$$Y_{im}^{i,b} = \sum_{t=0}^{T-1}\sum_{k=0}^{K-1} y_{i,b}^{(\ell)}(t,k)\sin\left(\hat{\phi}_i^{(\ell)}(t) + k\omega\right),$$

$$X_{re}^{i,b} = \sum_{t=0}^{T-1}\sum_{k=0}^{K-1} x_{i,b}^{(\ell)}(t,k)\cos\left(\hat{\phi}_i^{(\ell)}(t) + k\omega\right),$$

$$X_{im}^{i,b} = \sum_{t=0}^{T-1}\sum_{k=0}^{K-1} x_{i,b}^{(\ell)}(t,k)\sin\left(\hat{\phi}_i^{(\ell)}(t) + k\omega\right),$$

$$XY = \sum_{t=0}^{T-1}\sum_{k=0}^{K-1} \left(x_{i,b}^{(\ell)}(t,k)^2 + y_{i,b}^{(\ell)}(t,k)^2\right).$$

$$\hat{\rho}_{i,b+1}^{(\ell)} = \frac{1}{TK}\sqrt{\left[\left(Y_{im}^{i,b} + X_{re}^{i,b}\right)^2 + \left(Y_{re}^{i,b} - X_{im}^{i,b}\right)^2\right]}\Bigg|_{\hat{\theta}_{i,b+1}^{(\ell)}},$$

$$\hat{\psi}_{i,b+1}^{(\ell)} = \arctan\left(\frac{Y_{re}^{i,b} - X_{im}^{i,b}}{Y_{im}^{i,b} + X_{re}^{i,b}}\right)\Bigg|_{\hat{\theta}_{i,b+1}^{(\ell)}}. \tag{3.24}$$

The updated $\hat{\Theta}_{i,b+1}$ are approximated using numerical techniques and/or grid searches over L*H intervals. ML estimates of $\hat{P}_{i,b+1}$ and $\hat{\Psi}_{i,b+1}$ can be evaluated in closed form once the values of $\hat{\Theta}_{i,b+1}$ have been computed. This EM variant of the SKMA is much less computationally intensive than an L dimensional numerical technique or an L dimensional search involving approximately H^L/L intervals.

After a convergence criterion is satisfied, similar to Eqns. (3.7) and (3.13), the parameters $\left[\hat{P}_{i+1}, \hat{\Psi}_{i+1}, \hat{\Theta}_{i+1}\right] = \left[\hat{P}_{i,b+1}, \hat{\Psi}_{i,b+1}, \hat{\Theta}_{i,b+1}\right]$ are used in the next segmentation step.

Figure 3.1 shows schematically how the EM algorithm is used in conjunction with the SKMA.

The SKMA method and the suboptimal variant, described above, (denoted by SKMA - EM) are compared with the ML estimation method (denoted by ML) and described in Chapter 1, Section 1.4.

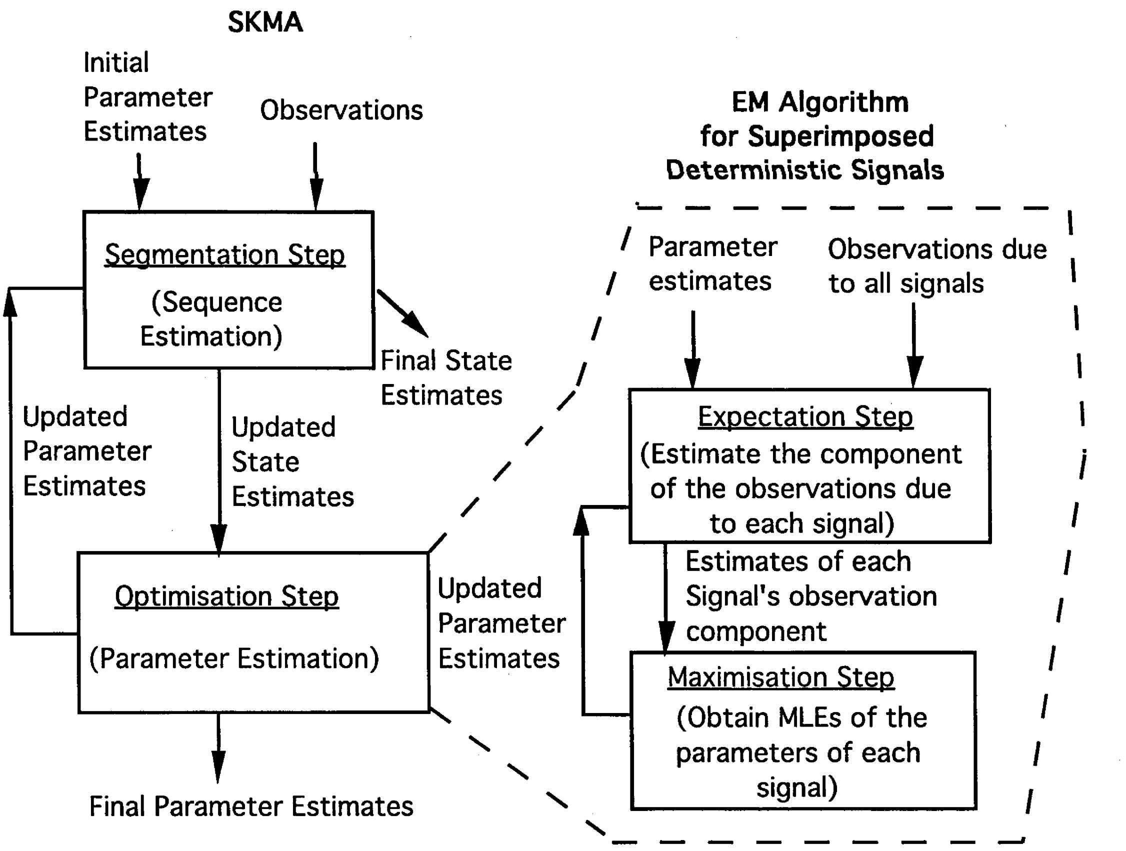

Figure 3.1: Joint State Sequence and Parameter Estimation using the SKMA and EM algorithms.

§3.4 Simulations and Results

The plane wave signal(s) are assumed to be convolutional coded QPSK, with constraint length 7 and rate ½. The generating polynomials, $G_0 = (1,0,1,1,0,1,1)$ and $G_1 = (1,1,1,1,0,0,1)$, are those found by Odenwalder as described in Sklar [1988]. The channels are assumed not to cause ISI, have the same baud rate and only a single sample per baud is used.

§3.4.1 Single Signal

A convolutional coded QPSK signal with block length 1000 symbols, amplitude $\rho = 1.0$, and phase offset $\psi = 0.0$, was generated. The initial state of the convolutional encoder was set to zero. WGN was added to obtain a 5dB SNR† signal. The SKMA was used to jointly demodulate the signal and estimate the signal's amplitude and phase offset.

Figures 3.2 and 3.3 show the trajectory of the combined amplitude and phase offset estimates. The real and imaginary values were determined by $\rho\cos(\psi)$ and $\rho\sin(\psi)$ respectively. The initial phase offset estimates were varied between $\pm\pi/2$ in increments of $\pi/20$ with initial amplitude estimates of 2.5 and 0.3 respectively. These plots show that there is a region, within approximately $\pm\pi/4$ of the true phase offset value, where the estimates converge‡ towards the true value. For initial estimates of the phase offset greater than $\pm\pi/4$ of the correct value, Figures 3.2 and 3.3 show many local maxima. These local maxima are not due to our convergence criteria as shown by simulations that were allowed to continue estimating the parameters beyond the convergence criteria. Further simulations showed that if the initial phase offset estimate was within approximately $\pm\pi/4$ of the true phase value, then the initial amplitude estimate could be at least as high as 10 times the true amplitude value and

† $\mathrm{SNR} = 10\log_{10}(\rho^2/\sigma^2)$, where ρ is the signal amplitude and σ^2 the noise variance.

‡ Convergence was determined to have been achieved when successive estimates of the amplitude and phase offset where within 10^{-4} of each other.

convergence towards the true value would still occur. This is because the signal model is linear and continuous in amplitude, and hence the amplitude estimate is less sensitive to incorrect initial conditions (this would not be the case for signals with discrete amplitude levels, as in quadrature amplitude modulation).

Figure 3.4 shows the estimate trajectories for 30 realisations all starting with initial estimates, $\rho_0 = 2.5$ and $\psi_0 = 0.4$. The plot shows that the amplitude and phase offset estimates have converged close to the correct value in just 2 iterations, the '+' symbols indicate the final ML estimate (MLE) for each realisation. For the 30 realisations, the average number of iterations until convergence was 3.5 and the average Bit Error Rate (BER) of the estimated signal (after convergence of the amplitude and phase offset had been achieved) was 0.64%. This is almost identical in performance to having exact knowledge of ρ and ψ, which gave an average BER of 0.63%.

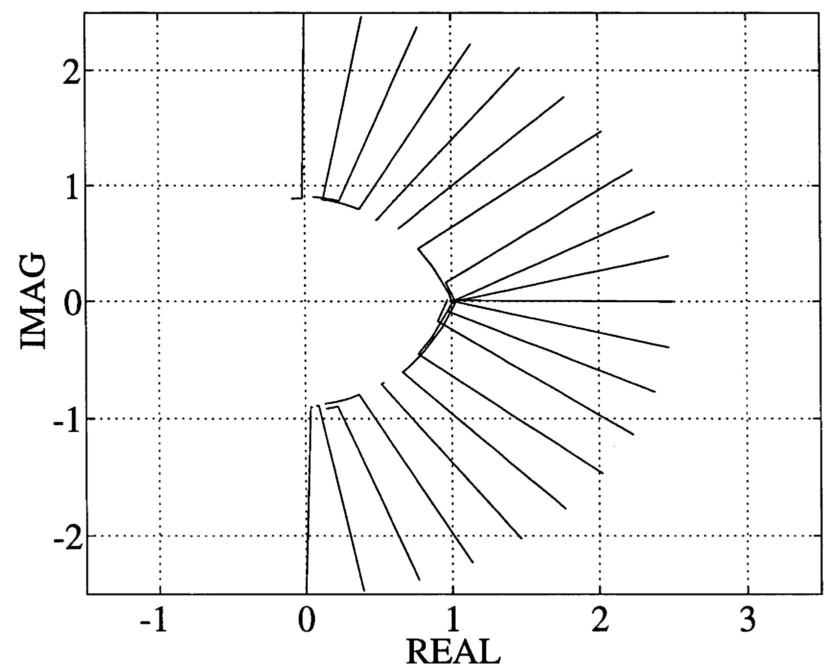

Figure 3.2: Estimation trajectory for initial phase estimates between $\pm\pi/2$ - $\rho_0 = 2.5$.

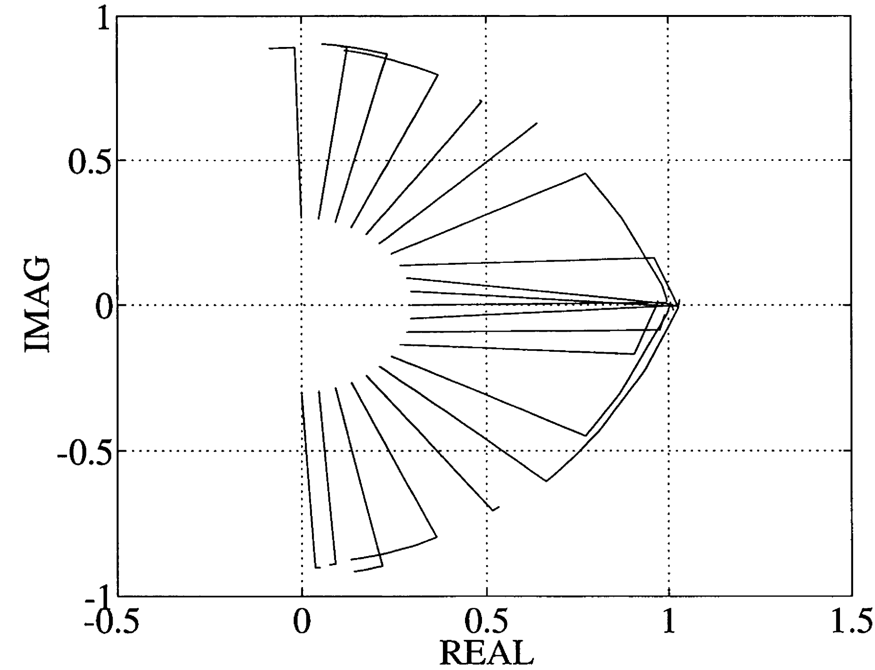

Figure 3.3: Estimation trajectory for initial phase estimates between $\pm\pi/2$ - $\rho_0 = 0.3$.

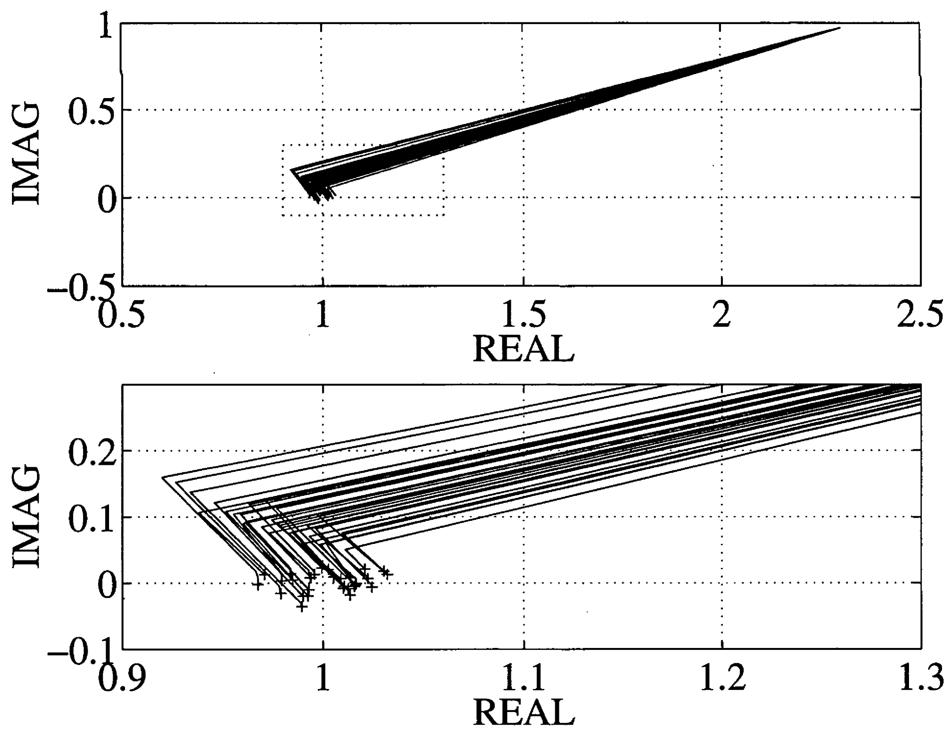

Figure 3.4: Estimation trajectories for 30 realisations - $\rho_0 = 2.5$ & $\psi_0 = 0.4$.

§3.4.2 Array Processing

§3.4.2.1 Single Signal

A single plane wave convolutional coded QPSK signal with block length 300 symbols, $\rho = 1.0$ and $\psi = 0.0$ was simulated to be incident, 23° from endfire, on a ULA of 5 sensors spaced one-half wavelength.

Figures 3.5 and 3.9 show the performance of the VA verse the AOA estimate used in the algorithm, ρ and ψ were set to their true values. These plots are for 20dB and 0dB per sensor signals respectively. The plots show a region within approximately ±13° of the true AOA for which the VA decodes the signal correctly. These plots show that the VA is robust to small errors in AOA estimates.

The SKMA is now used to jointly demodulate the signal and estimate the AOA of the signal (ρ and ψ are assumed known and not estimated).

The SKMA is first examined using a 20dB per sensor signal. Figure 3.6 shows the likelihood surface (Eqn. (3.10)) obtained in the first iteration of the SKMA, for all initial AOA estimates (i.e. 0° - 180°). The small likelihood values in this plot correspond to the AOA estimate areas in figure 3.5 which were not decoded correctly. Figure 3.7 shows the likelihood surface obtained in the final iteration of the SKMA, again for all initial AOA estimates (0° - 180°). The plot shows that the likelihood surface has been smoothed with successive iterations of the SKMA. It also shows a continuous ridge of high likelihood values at the true AOA.

Figure 3.8 shows the final AOA estimate of the SKMA verse initial AOA estimate (solid line) and the number of iterations the SKMA took to converge to this value (dashed line). Note how (at 20dB) the SKMA correctly estimates the signal's AOA regardless of the initial AOA estimate. This is explained by examining figure 3.7 and noting the continuous ridge of high likelihood values at the signal's true AOA. Note also that the initial AOA estimate areas in which the number of iterations required to converge increases, these areas corresponds to the AOA estimate areas in figure 3.5 in which the VA does not decode the signal correctly.

Figures 3.10, 3.11 and 3.12 are for a 0dB signal and correspond to figures 3.6, 3.7 and 3.8 for the 20dB signal case, respectively. In figure 3.10 the initial AOA estimate region in which the likelihood values are small corresponds to the AOA estimate region in figure 3.9 where the VA does not decode the signal correctly, this is the similar to what occurred in the 20dB signal case. Note how the final likelihood surface (figure 3.11 - 0dB case) contains a relatively flat region where the likelihood values are small, this corresponds to the initial AOA estimate region in figure 3.12 where the AOA estimate obtained from the SKMA is incorrect. Note also the increasing number of iterations which occurs either side of this region and the sharp change which occurs (both in the number of iterations - figure 3.12, and in the likelihood surface - figure 3.11) when the SKMA does not converge to the correct AOA estimate.

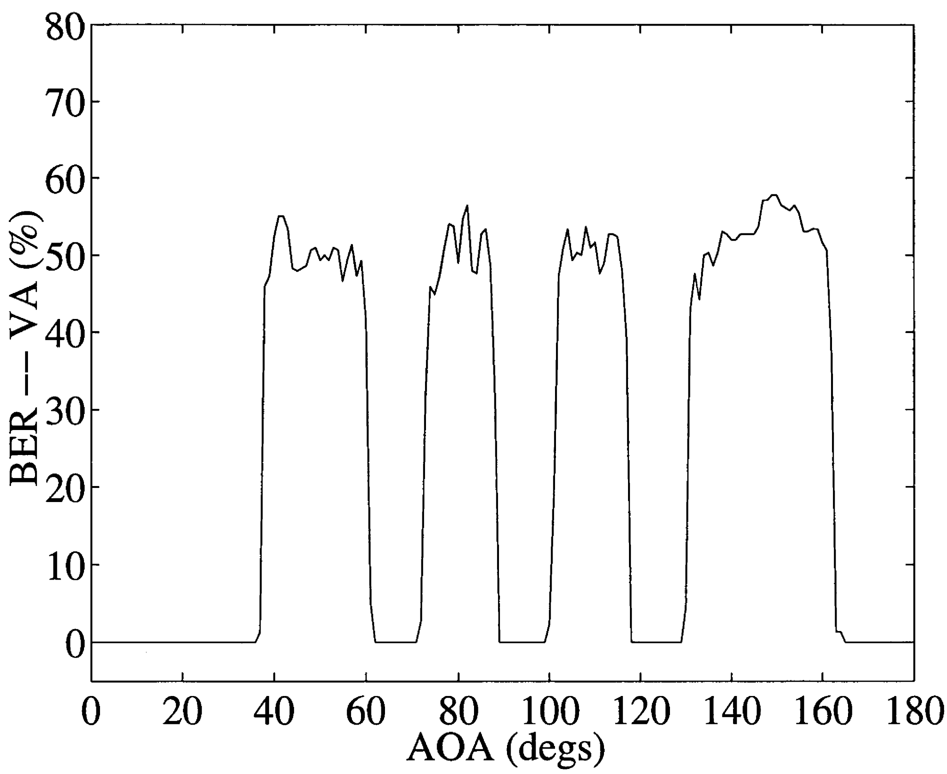

Figure 3.5: Performance of the VA verse AOA - 20dB/Sensor Signal.

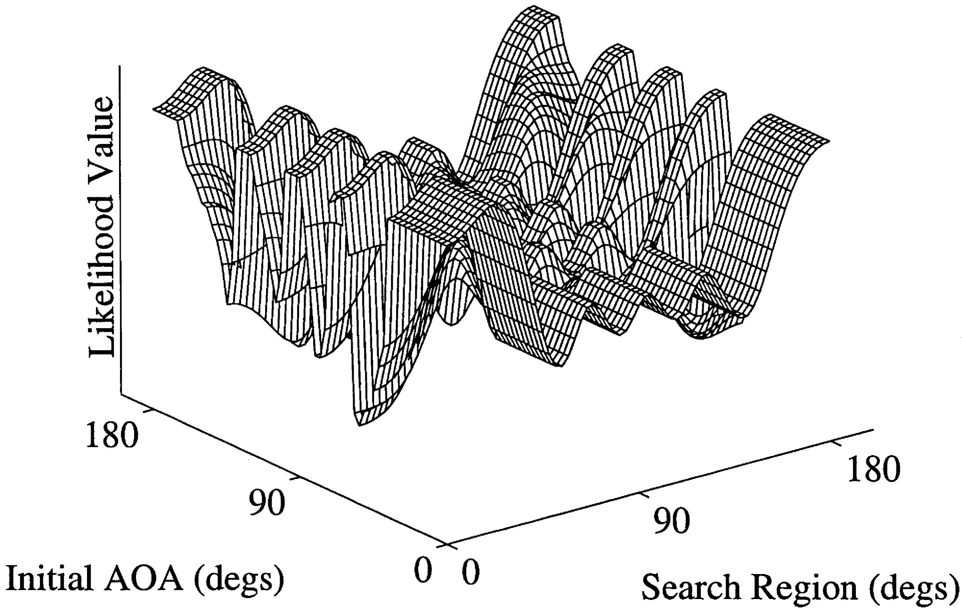

Figure 3.6: Likelihood function surface, Initial AOA estimate vs Search Region, First Iteration - 20dB/Sensor signal.

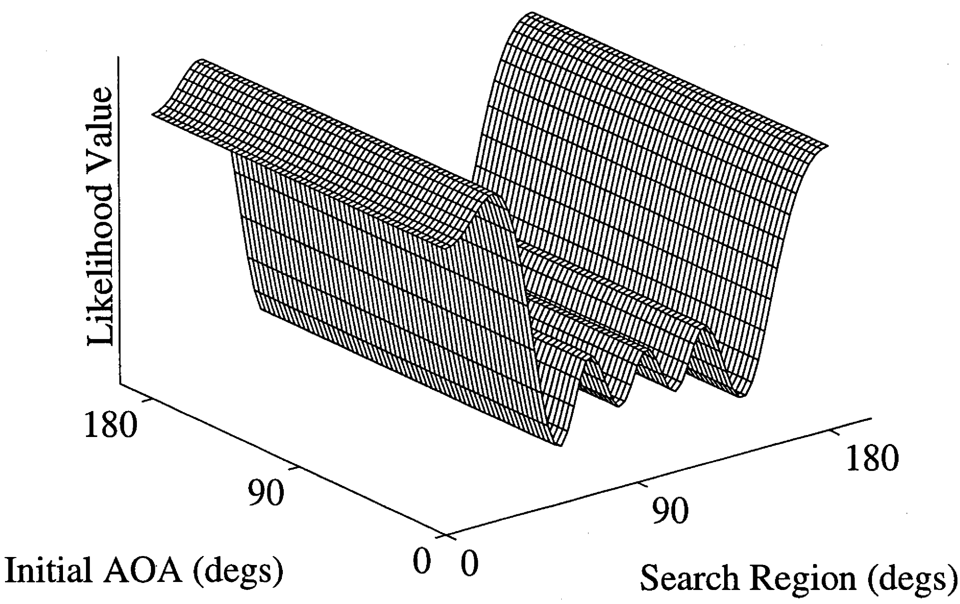

Figure 3.7: Likelihood function surface, Initial AOA estimate vs Search Region, Final Iteration - 20dB/Sensor signal.

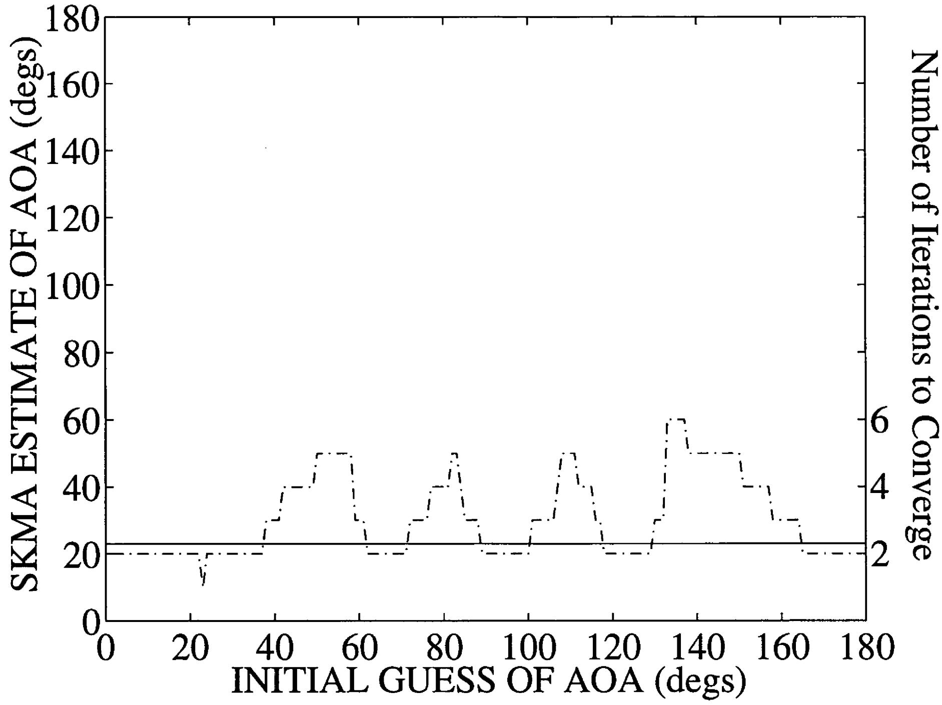

Figure 3.8: Final AOA estimate vs Initial Guess AOA (solid line) with Number of Iterations of the SKMA to converge (dashed line) - 20dB/Sensor signal.

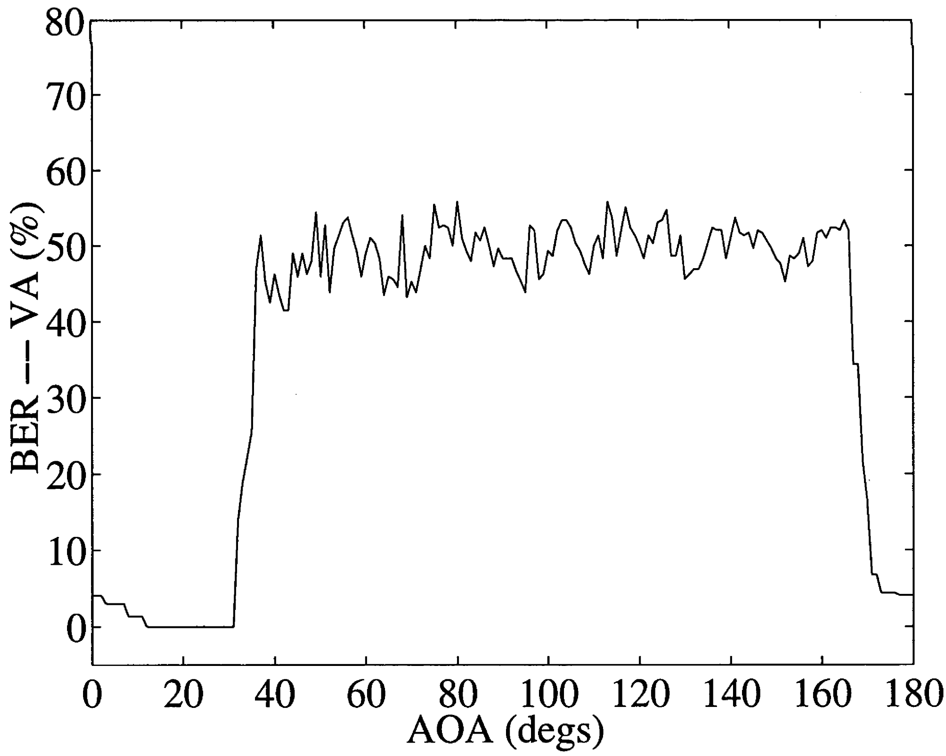

Figure 3.9: Performance of the VA verse AOA - 0dB/Sensor signal.

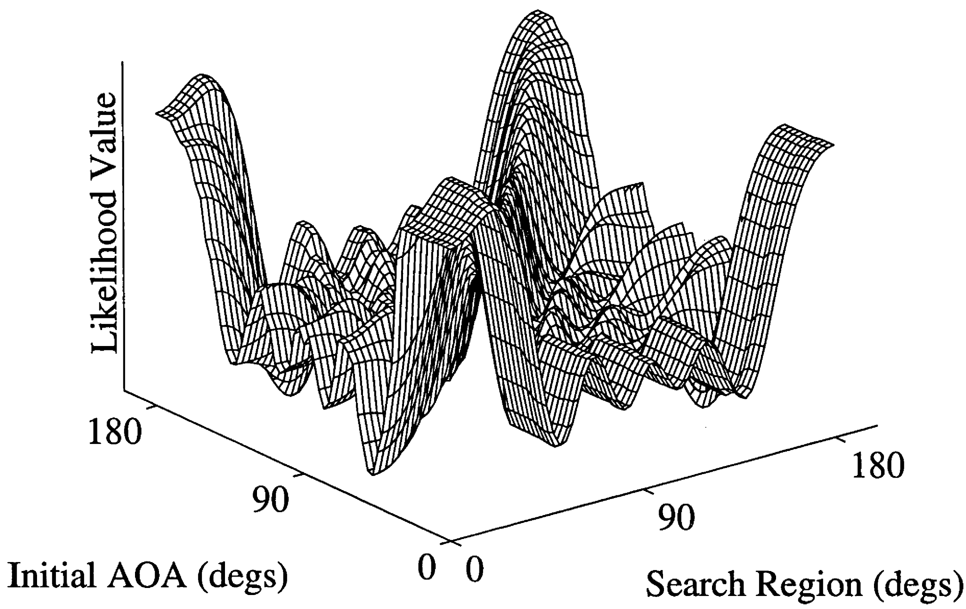

Figure 3.10: Likelihood function surface, Initial AOA estimate vs Search Region, First Iteration - 0dB/Sensor signal.

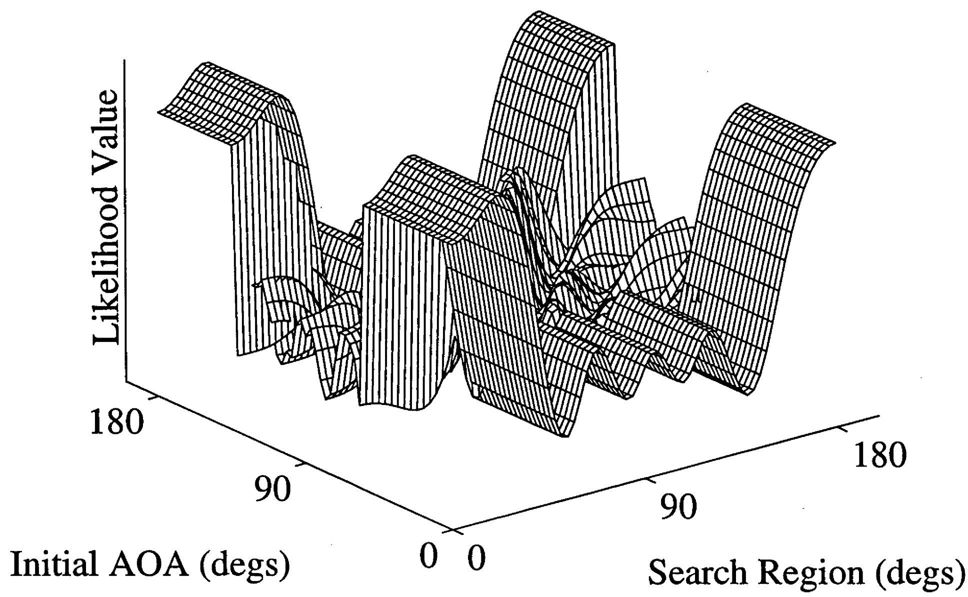

Figure 3.11: Likelihood function surface, Initial AOA estimate vs Search Region, Final Iteration - 0dB/Sensor signal.

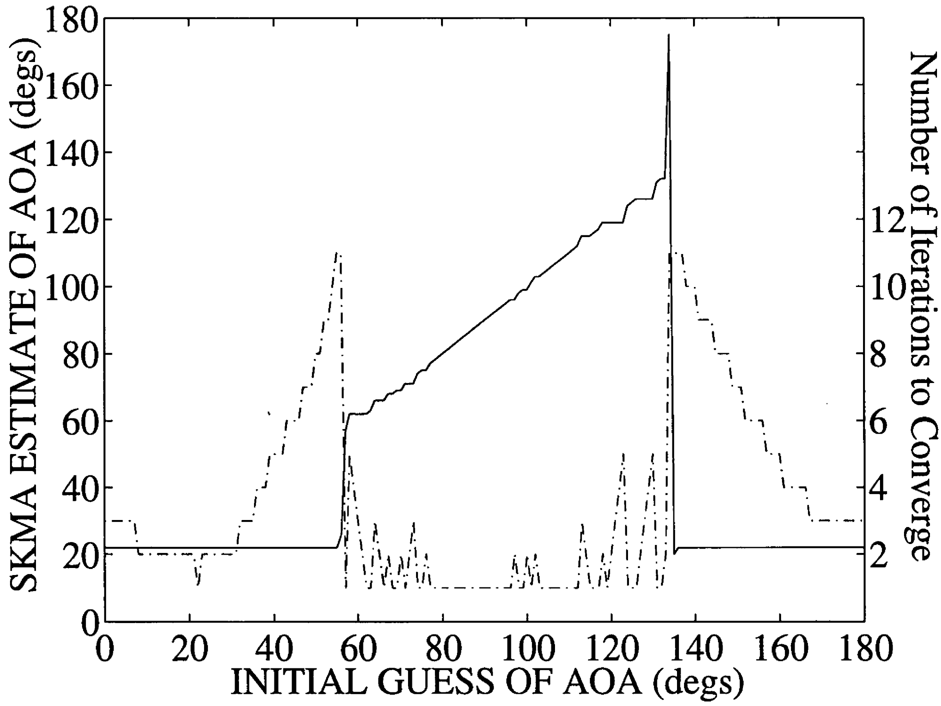

Figure 3.12: Final AOA estimate vs Initial Guess AOA (solid line) with Number of Iterations of the SKMA to converge (dashed line) - 0dB/Sensor signal.

Figure 3.13 compares the Root Mean Square (RMS) error of estimating the AOA using the SKMA and the ML method for one signal. This comparison is done over 300 realisations per SNR value for each method. The signal was simulated using the same parameters as above. The initial estimate of the AOA for the SKMA was randomly chosen (using a uniform distribution) in the range 10° to 36°, and the search used by the ML method was also constrained to this region. The search grid (for both methods) was spaced 0.2°. A small improvement over the ML method is noted.

In figure 3.14 the variances of the estimates obtained using the SKMA and ML methods are compared to a Cramer Rao Bound (CRB). The CRB used is for a single deterministic signal observed in Gaussian noise and was calculated using the equation given in Haykin [1991, page 277] and has been labelled the ML CRB. Although the variance of the SKMA estimates are better than the ML CRB, it should be noted that the ML CRB does not take into account the signal model. (Note no bounds has been determined for the SKMA estimation methods in this thesis, as it is beyond the scope of this work, see Chapter 6, Section 6.3.)

Figure 3.15 shows a comparison between the percentage BER obtained using the two methods. The BER for the SKMA is determined from the final demodulated message sequence obtained using the algorithm. The BER for the ML method is determined by first obtaining the ML estimate for the AOA and then using this estimate to determine the estimated state sequence using the VA and hence the demodulated message sequence (see Chapter 1, Section 1.4). As shown in the figure, there is no difference in the BER between the two methods. This can be explained by examining figures 3.5 and 3.9 which show that the VA is robust to small errors in the AOA estimate.

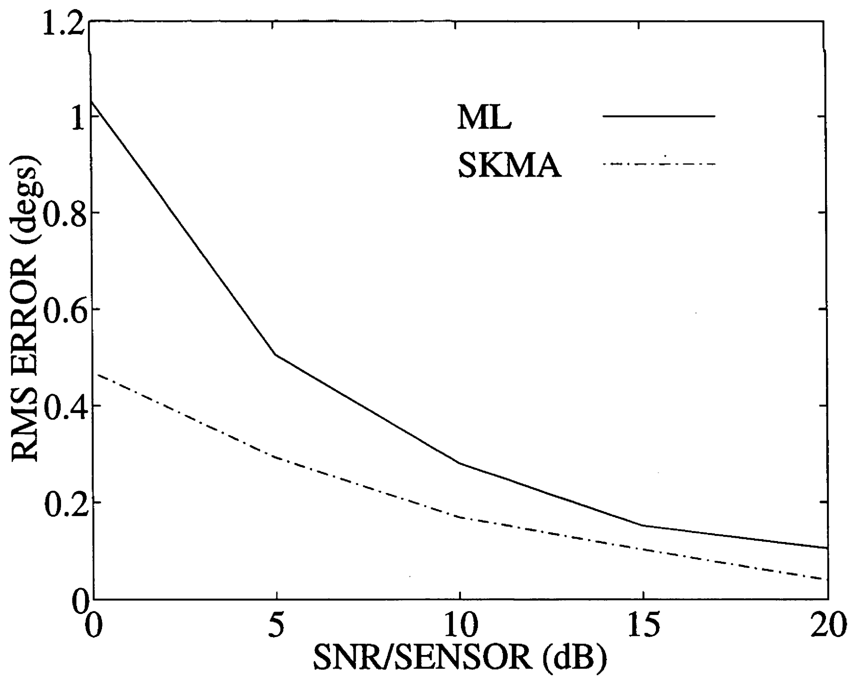

Figure 3.13: AOA estimation for one signal ML vs SKMA methods.

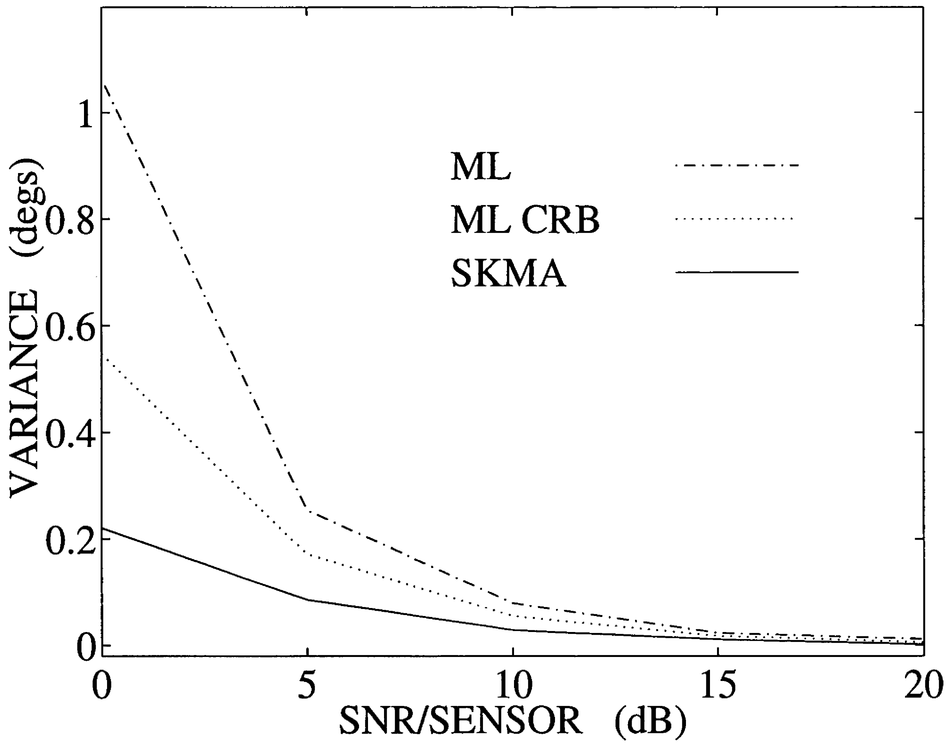

Figure 3.14: AOA estimate variance for one signal ML vs SKMA methods.

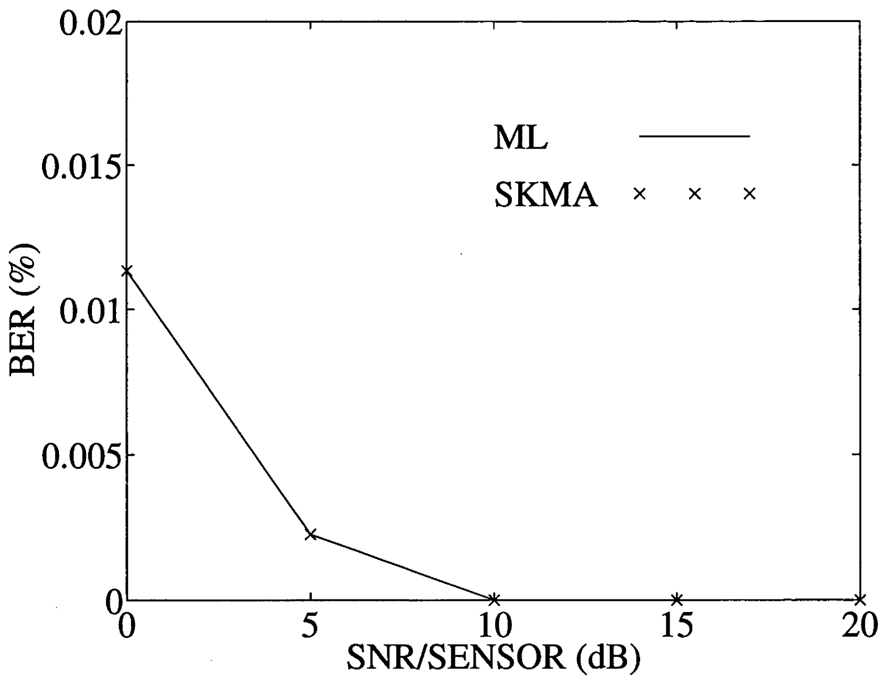

Figure 3.15: BER for one signal ML vs SKMA methods.

The SKMA relies on the signal model and iterative estimation in order to obtain the improved parameter estimates compared to the ML method (which is a sequential estimation scheme that does not take into account the signal model, see Chapter 1, Section 1.4). So, how robust is the SKMA to incorrect model assumptions?

Figure 3.16 examines the accuracy of the AOA estimates obtain from the SKMA when the received signal is a tone and not a convolutional coded signal (as assumed in the SKMA's signal model). The results from this simulation are compared to the case when the signal is a convolutional coded signal (labelled as "CC SIGNAL") which is correctly modelled by the SKMA. This plot shows that the SKMA is robust to this type of incorrect model assumption, as the RMS error for the AOA estimates for both the convolutional coded signal and the tone are very similar. The AOA estimates for the tone being only slightly worst than for convolutional coded signal. This can be explained by the fact that a tone signal is equivalent to a convolutional coded signal which has all zero input bits (which cause all the $\hat{s}_i(t)$s obtained from Eqn (3.8) to be zero), and therefore is not really an incorrect model assumption even though it may appear to be.

Figure 3.17 examines an incorrect model assumption which involves the received signal not having the same generating polynomial as is expected in the SKMA's signal model. The received signal was generated with generating polynomials, $G_0 = (1,0,1,1,0,0,1)$ and $G_1 = (1,1,1,0,0,0,1)$, instead of the standard generating polynomials (i.e. $G_0 = (1,0,1,1,0,1,1)$ and $G_1 = (1,1,1,1,0,0,1)$). The received signal's generating polynomials may have resulted due to hardware failure in the signal generator. This plot shows the RMS error when estimating the AOA of both a signal with the incorrect generating polynomials (labelled "BROKEN GENS") and a signal with the correct generating polynomials (labelled "CC SIGNAL"). This plot shows that the SKMA is not robust to this type of incorrect model assumption.

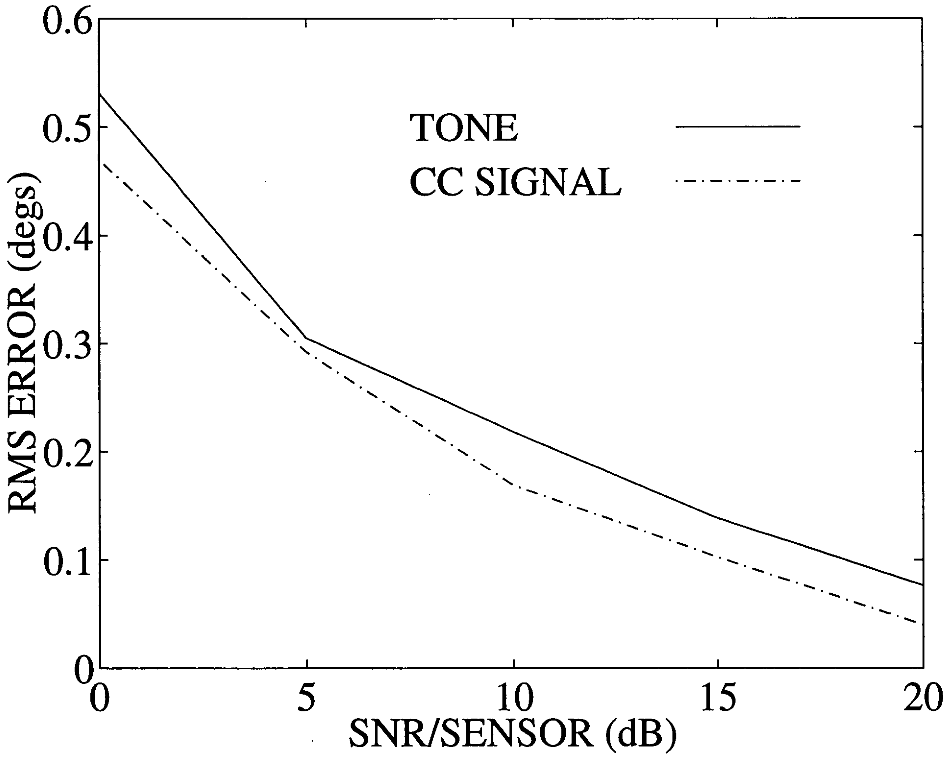

Figure 3.16: RMS Error for CC signal vs tone using the SKMA method.

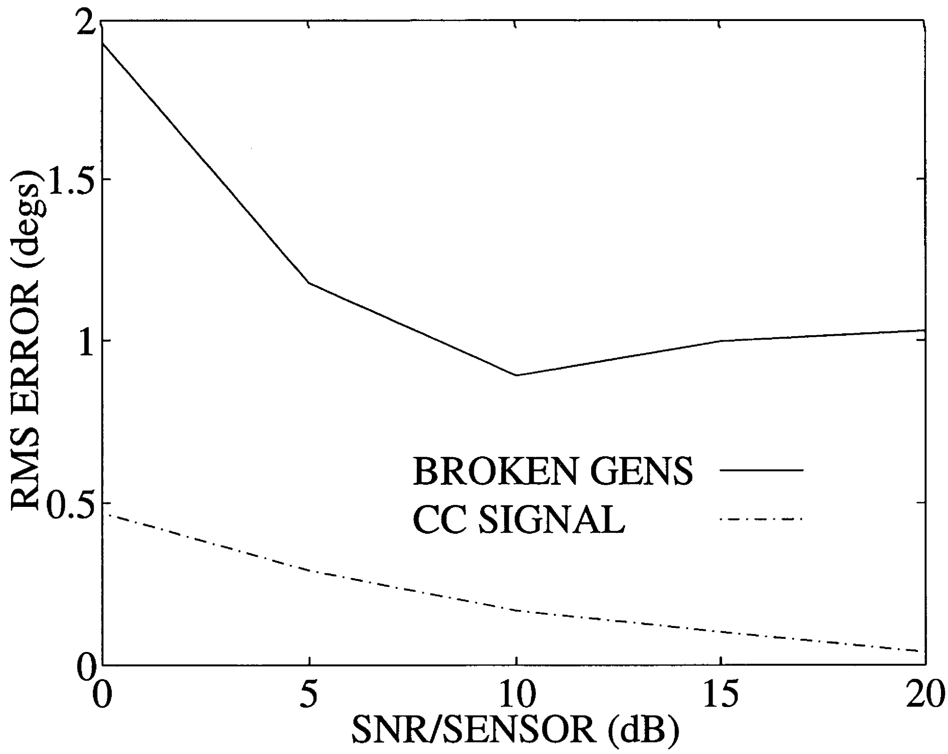

Figure 3.17: RMS Error for CC signal vs Broken signal using the SKMA method.

§3.4.2.2 Superimposed Signals

Two plane wave convolutional coded QPSK signals with block length 300 bits were simulated to be incident on a uniform linear array of 5 sensors spaced one-half wavelength. The two signals both have $\rho = 1.0$ and $\psi = 0.0$ (which are again assumed known and not estimated) and the same generating polynomials (given at the beginning of Section 3.4). The signals have statistically independent input message sequences, are incident on the array at 23° and 28° from endfire, and are demodulated using 3 methods: the ML method (Chapter 1, Section 1,4); the SKMA; and SKMA - EM described in this chapter.

Figure 3.18 shows the RMS error of estimating the AOA for two signals using the three methods. These results are calculated for 300 realisations per SNR value for each method. The initial estimate of the AOA for each signal was randomly chosen (using a uniform distribution) in the range 13° to 38° and the search used by the ML method was also constrained to this region. The search grid for all methods was again spaced 0.2° giving H = 126 values. The SKMA and the SKMA - EM methods are shown to be significantly more accurate at estimating the AOAs when compared to the ML method, particularly at low SNR. This plot also illustrated that there is little different between the accuracy of the SKMA and the SKMA - EM methods. Computationally there is a large difference between these two methods. The following discussion explains why.

In this example using the SKMA method to estimate the AOAs, Eqn. (3.18) was solved using a grid search involving $H^2/2 \approx 8000$ intervals for each iteration of the algorithm. An average number of approximately two iterations occurred until convergence was achieved (for the 5 dB case). This means that Eqn. (3.18) is calculated approximately 16000 times during an average realisation which is very computationally intensive. In using the SKMA - EM method for estimating the AOAs Eqn. (3.23) was also solved using a grid search, however it was only over 2*H

= 252 intervals for each iteration of the EM part of the method. An average number of approximately eight iterations occurred until convergence was achieved (for the 5 dB case), thus Eqn. (3.23) was calculated approximately 2000 times. Eqn. (3.20) was calculated on average approximately 32 times. Clearly, the SKMA - EM method is significantly less computationally intensive than the SKMA method.

The two signals can be resolved if both signals' AOA estimates satisfy the following condition [Pillai 1989]:

$$\left|\theta^{(\ell)} - \hat{\theta}^{(\ell)}\right| < \alpha \quad \ell = 1,2 \tag{3.25}$$

where $\theta^{(\ell)}$ is the true AOA for signal ℓ,

$\hat{\theta}^{(\ell)}$ is the estimated AOA for signal ℓ,

$$\alpha = \frac{\theta^{(1)} + \theta^{(2)}}{2}.$$

Figure 3.19 shows the probability of resolving the signals for the three methods considered. It is easily seen that the signal resolution is far superior for the SKMA and SKMA - EM methods compared to the ML method at the SNR values shown, with negligible difference between the SKMA and SKMA - EM methods.

Figure 3.20 shows the average BER for the two signals as determined using the three methods. The BERs obtained using SKMA and SKMA - EM methods are almost identical. These two methods show a remarkable threshold extension (~20dB) over the deterministic ML method.

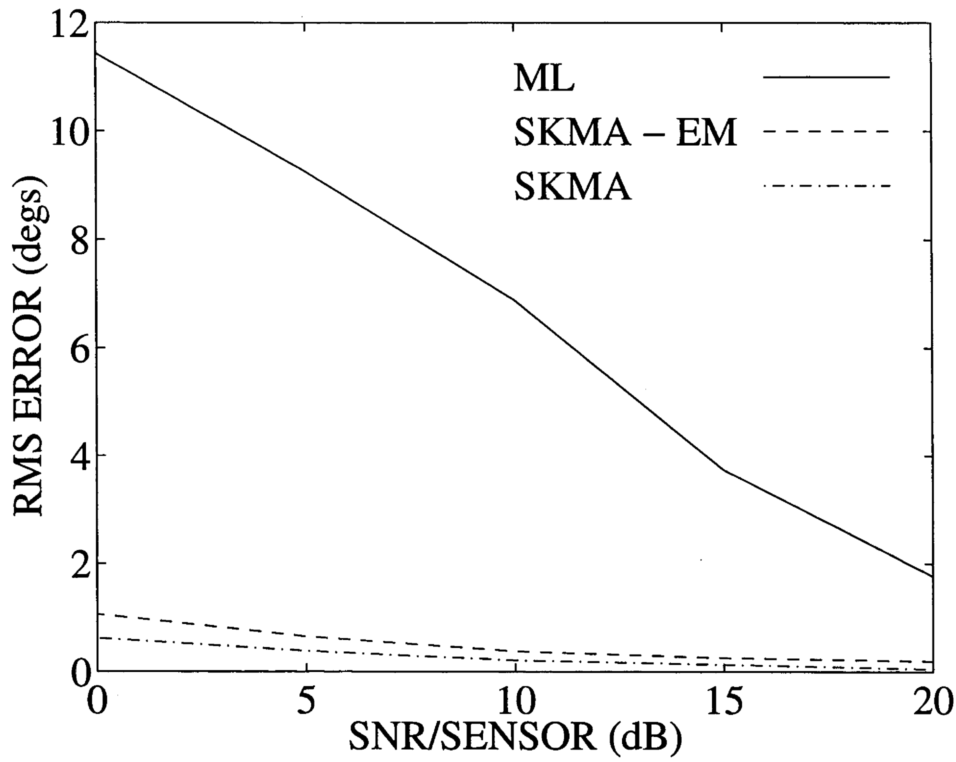

Figure 3.18: AOA estimation for two signals ML vs SKMA vs SKMA-EM methods.

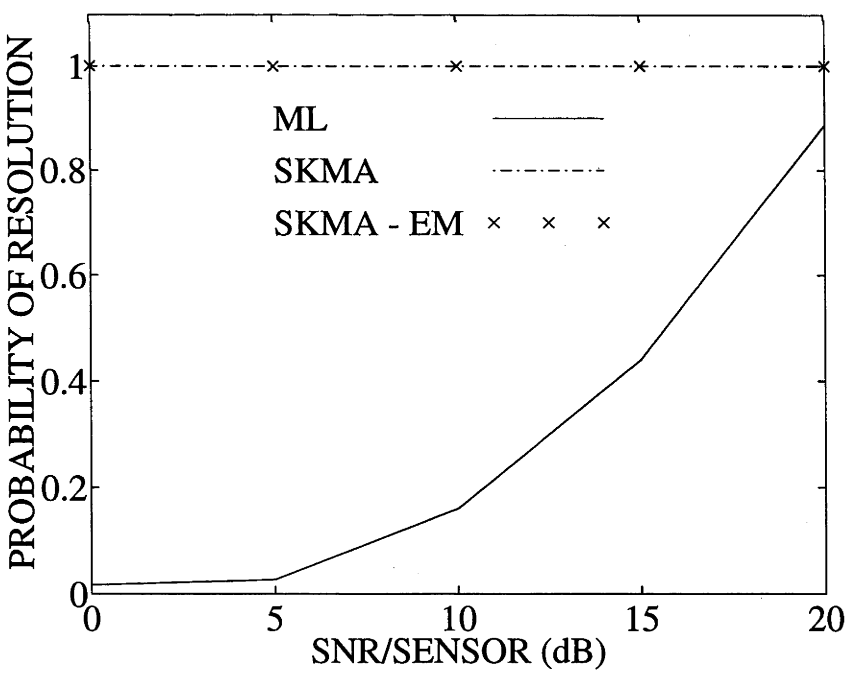

Figure 3.19: Probability of resolution for two signals: ML vs SKMA vs SKMA - EM methods.

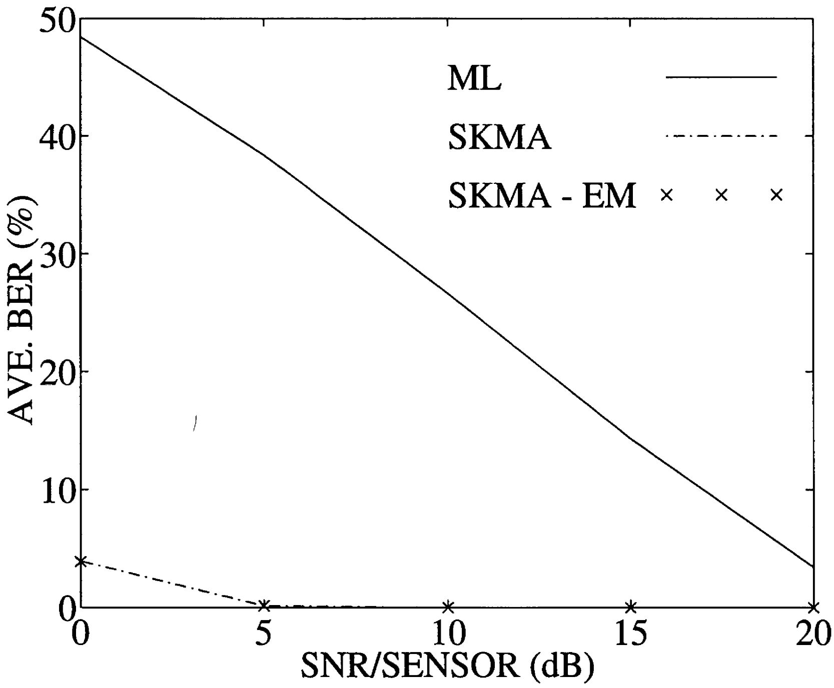

Figure 3.20: Average BER for two signals ML vs SKMA vs SKMA-EM methods.

Examining how robust the SKMA- EM method is to an incorrect model assumption, one signal is generated with the same incorrect generating polynomials as in Section 3.4.2.1, while the other signal is generated with the correct generating polynomials, all other details of the two signals are the same as those at the beginning of this section.

Figure 3.21 shows the RMS error of estimating the AOA of the two signals for both the ML method and the SKMA - EM method. This plot shows that the ML results are the same as those obtained in figure 3.18, which is to be expected since the ML method does not use any signal model information. The SKMA - EM method's performance is worse than when the signal model is correct (see figure 3.18), but at low SNR is significantly better than the ML method.

Figure 3.22 shows the ML and SKMA - EM methods' probability of resolving the two signals. It is easily seen that the signal resolution of the SKMA - EM method is far superior to the ML method at the SNR values shown.

Figure 3.23 shows the BER obtained by both the ML and SKMA - EM methods for the signal which has the same generating polynomials as expected by the methods. The signal which has the broken generating polynomials will not be demodulated correctly anyway and therefore the BER for this signal is not shown. This plot illustrates that even though only one signal is incorrectly modelled by the SKMA - EM, it affects the demodulation of the signal correctly modelled, while the ML method's performance is basically unchanged – as expected (see figure 3.20 which shows the average BER for two signals).

Figures 3.21 and 3.22 show that in estimating AOA, the SKMA - EM method is robust to the incorrect signal model introduced. This can be explained by the fact that even though one signal is incorrectly modelled (in this case the generating polynomials), the signal was still a QPSK modulated signal which was modelled correctly and this is what improved the AOA estimates over the ML method. However if the signal modulation model is incorrect then it is expect that the SKMA - EM would not be robust in estimating the AOAs. Figure 3.23 shows that in demodulating the signals, the SKMA - EM is not robust to an incorrect signal model involving the generating polynomials. This caused the signal correctly modelled to be demodulated incorrectly and worse than the ML approach at high SNR.

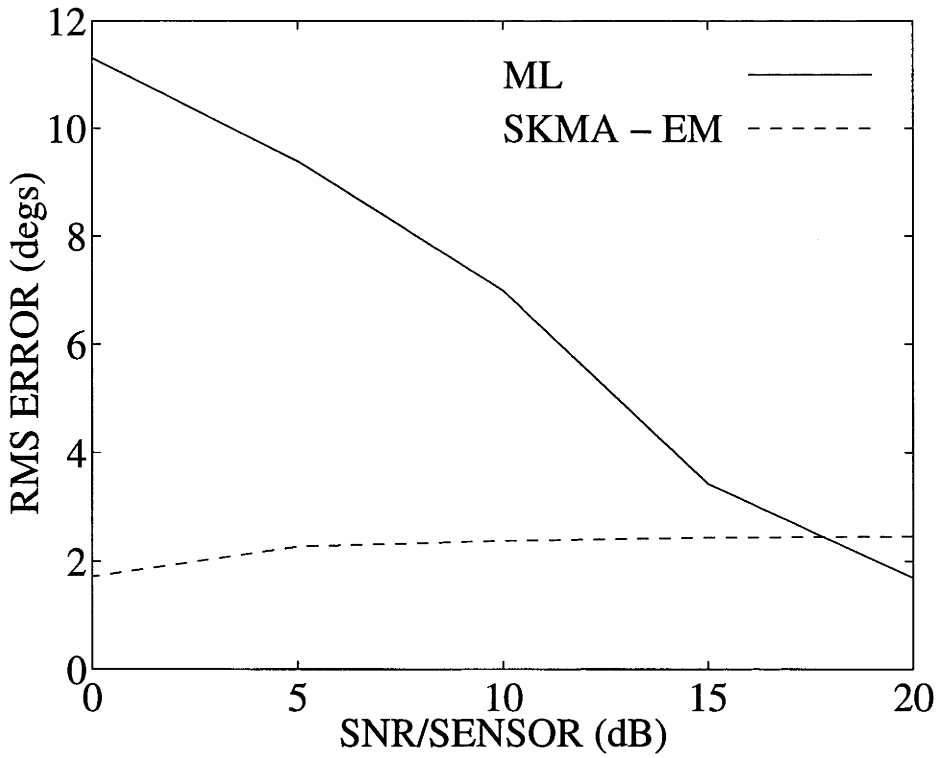

Figure 3.21: AOA estimation for one signal and one broken - ML vs SKMA - EM methods.

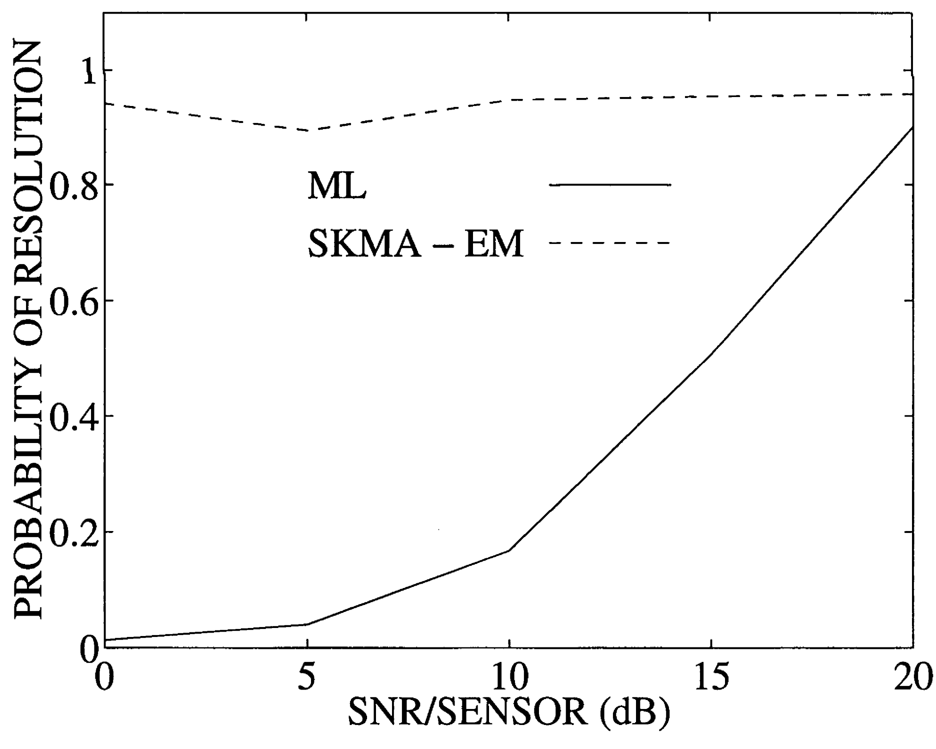

Figure 3.22: Probability of resolution for two signals, when estimating two signals but one is broken - ML vs SKMA - EM methods.

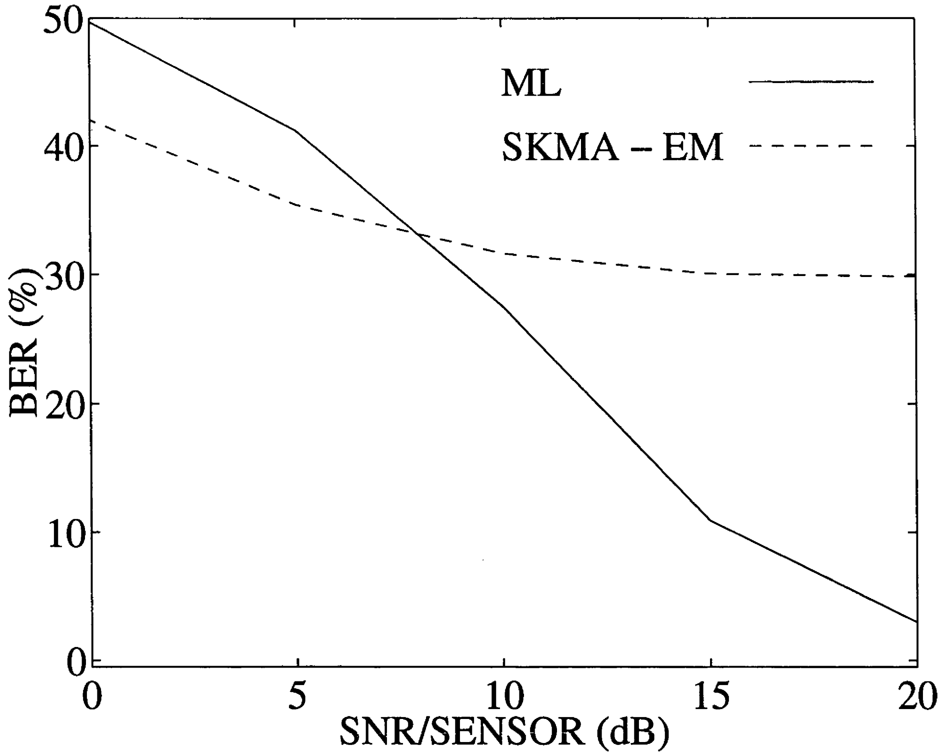

Figure 3.23: BER for one signal, when estimating two signals but one is broken - ML vs SKMA - EM methods.

§3.5 Conclusion

In this chapter a method for spatial filtering superimposed convolutional coded QPSK signals, using the SKMA was developed. A version which reduces the computational complexity in the parameter estimation section of the SKMA method was also developed (called the SKMA - EM method). These methods were compared with the ML method described in Chapter 1, Section 1.4. For the examples discussed, it was shown that the SKMA and SKMA-EM methods are significantly more accurate than the ML method in demodulating the signals and in the estimation of the AOAs, particularly at low SNR. However, the improvements are achieved through an increase in the computational complexity of the problem and hence the processing time for obtaining solutions is also increased. The SKMA - EM method significantly reduced the computational complexity involved in estimating the parameters without any significant loss of performance in the demodulation of the signals or in the estimation of the AOAs. The improved accuracy of the SKMA and SKMA - EM methods provides a remarkable threshold extension, ~20dB, compared to the ML method. This improvement is due to jointly estimating the signals' state sequences and parameters and also that the signal models were taken into account (i.e. the discrete nature of the phase of the transmitted signals and the Markovian nature of the state sequences). This was illustrated in the simulations which examined how robust the SKMA and SKMA - EM methods are to an incorrect signal model involving the generating polynomials.

Chapter 4

Reduced Complexity Computation

§4.1 Introduction

This chapter develops a reduced complexity on-line state sequence and parameter estimator for superimposed convolutional coded signals. Joint state sequence and parameter estimation is achieved by iteratively estimating the state sequence via a reduced complexity Viterbi algorithm (RCVA) and the signal parameters via a recursive (on-line) EM algorithm.

Chapter 3 developed a method of jointly estimating the state sequence and parameter estimates of superimposed convolutional coded signals. The method used was a HMM based technique, the SKMA. There are two main computational problems with this method, these are:

1) Estimating the state sequences of superimposed signals.

2) Estimating the signal parameters, some computational complexity reduction was achieved by using the EM algorithm for superimposed deterministic signals in the SKMA's optimisation (parameter estimation) step.

For one signal the VA computes and retains probabilities and path information for $F = 2^N$ states at each time sample. When there are L superimposed signals present the Viterbi decoder must be able to handle F^L states. Therefore the Viterbi decoder for the superimposed signals problem requires extensive storage which is exponential in the constraint length of the code [Sklar 1988] and the number of signals, and linear in the decoding block size.

Both methods however are still computationally intensive. The reduced complexity on-line state sequence and parameter estimator reduces the computational burden of the SKMA by using:

1) the RCVA (developed in Section 4.2) in an on-line mode in the segmentation (state sequence estimation) step.

2) a recursive EM algorithm in the optimisation (parameter estimation) step.

There are many algorithms which have been devised to reduce the complexity of sequence estimation [Anderson and Mohan 1984, Eyuboğlu and Qureshi 1988, Duel-Hallen and Heegard 1989]. Anderson and Mohan [1984] surveyed a number of sequential coding algorithms, performed a cost analysis on each and compared them with the VA. The algorithms surveyed were: The Single Stack, The Fano, The Two-Cycle, The Stack, The Bucket and The M-Algorithm. Eyuboğlu and Qureshi [1988] and Duel-Hallen and Heegard [1989] developed similar algorithms which perform Reduced State Sequence Estimation (RSSE) for ISI channels. Eyuboğlu and Qureshi's [1988] algorithm uses set partitioning and is applicable to finite ISI channels, while Duel-Hallen and Heegard's [1989] algorithm can handle infinite ISI channels, that is recursive channels with infinite impulse response. Sheen and Stüber [1992 and 1993] give a method of determining the error bounds for RSSE. The RSSE algorithms of Eyuboğlu and Qureshi [1988] and Duel-Hallen and Heegard [1989] use a parameter which determines the number of reduced states to be used in the decoding and hence with appropriate selection of this parameter, optimal Viterbi decoding can still be achieved. Anderson and Offer [1994] recently wrote "that no commonly tabulated good codes benefit from reduced state sequence detection", however in this chapter it is shown that the RCVA which is developed, can be used for decoding superimposed convolutional coded signals.

In the case where the input message bits of the superimposed signals are i.i.d., reduced complexity Viterbi decoding becomes more attractive than the algorithms surveyed by Anderson and Mohan [1984], since in situations where the environment is changing, the number of reduced states may be adaptively varied in order to acquire and maintain lock onto the correct state sequence.

Simulations are used to show the improved performance of the on-line RCVA when compared to an on-line version of the standard RSSE technique. The performance of on-line versions of the RSSE, RCVA and VA are also compared when they are used to estimate the state sequence as part of the reduced complexity on-line state sequence and parameter estimator. These simulations show that the estimator using the RCVA improves the probability of locking onto (acquiring) the correct state sequence and parameter estimates. When lock is acquired it is on average faster than the estimator using the standard RSSE technique, while only performing a little worse than the estimator using the full state VA.

This chapter is organised as follows: In Section 4.2 the RCVA for superimposed signals is developed. Section 4.3 describes the reduced complexity joint state sequence and parameter estimator for superimposed convolutional coded signals and details the on-line EM algorithm for superimposed deterministic signals. Simulations are presented in section 4.4, with some conclusions presented in section 4.5.

§4.2 Reduced Complexity Viterbi Algorithm

In this section, the reduced complexity Viterbi algorithm for state sequence estimation is developed. The signal models are the same as those used in Chapter 3. The parameter vector $\Theta_\ell(t)$ is assumed known. Section 4.3 describes an on-line joint state sequence and parameter estimation method (which reduces the computational complexity of the block processing method described in Chapter 3) for when $\Theta_\ell(t)$ is unknown and varying.

An estimate of the state sequence $\{\hat{s}(t)\}$, and hence the message bits $\{\hat{b}(t)\}$, can be obtained from the received observations using the VA, for some block length T. Practical full state (optimal) Viterbi decoders currently exist for signals which have constraint lengths of $N \leq 10$ [Sklar 1988]. For sufficiently high SNR the signal can be decoded using an algorithm of reduced complexity compared to the optimal VA [Anderson and Mohan 1984, Eyuboğlu and Qureshi 1988, Duel-Hallen and Heegard 1989].

The RSSE algorithms developed by Eyuboğlu and Qureshi [1988] and Duel-Hallen and Heegard [1989] work well for ISI problems where there are many states and many observations. A convolutional encoder maps many states into few observations and in this case the RSSE algorithms can not be relied on to work successfully. The RSSE algorithms do however, give more flexibility when low SNR signals are encounted, when multiple signals are to be decoded, or in changing environments, since these algorithms can easily revert back to full state optimal algorithms.

A RSSE is constructed by partitioning the convolutional code's shift register state $s(t)$, Eqn. (1.2), using μ the reduced memory of the code (see Duel-Hallen and Heegard [1989]), as follows:

$$s(t) = [r(t), p(t)] \tag{4.1}$$

$$\text{where} \quad r(t) = [b(t-N+1), \ldots, b(t-\mu-1)],$$

$$p(t) = [b(t-\mu), \ldots, b(t)], \text{ and}$$

$$0 \leq \mu < N.$$

Note that when μ = N-1, $p(t) = s(t)$ and the RSSE is equivalent to the full state VA and consists of 2^N states. For $\mu <$ N-1 the complexity of the RSSE reduces to a fixed number of $2^{\mu+1}$ states. Note that r(t) is obtained from s(t-1) and at time t, the RSSE only searches over the states ranging from [r(t), 0, ..., 0] to [r(t), 1, ..., 1] (i.e. the elements of p(t) are set to all zeroes and all ones respectively).

The RSSE is used as the basis for the development of the RCVA. A description of the full state VA can be found in Rabiner's [1989] tutorial paper on HMMs. Using Rabiner's notation (replacing q_t^* with $\hat{s}(t)$), and the RSSE partitioning, the RCVA procedure for finding the best state sequence is now described. The algorithm is presented in an on-line mode so as to reduce the storage requirements which grow linearly with the signal length [Sklar 1988]. Note that in the following description the binary representation of a state or its corresponding decimal integer value are interchanged according to the context in which it is used.

1) Initialise:

Let T be the block length, and h the delay after which a state doesn't affect the observations (h << T)†.

Choose a $\hat{s}(0)$, the state estimate at t = 0 (guess if no other information is available to assist the choice) and initialise π, the initial state distribution probability vector (uniformly if no other information is available).

Based on $\hat{s}(0)$ obtain r(1) (see Eqn. 4.1) and thus obtain upper and lower limits as follows:

$$\text{Lower Limit; } \mathrm{Lo}(1) = [\mathrm{r}(1), \mathrm{p}(1) = 0]^{\ddagger}$$
$$\text{Upper Limit; } \mathrm{Up}(1) = [\mathrm{r}(1), \mathrm{p}(1) = 1]$$

These limits are used to control the range of states for which the following calculations are performed. An interval over p(t) is searched since after a number of iterations the elements of r(t) are the result of successive

† A "h" of 4 or 5 times the constraint length of the code is sufficient [Heller and Jacobs 1971].
‡ Where p(1) = 0 means that each element of p(1) is set to zero.

estimates and therefore are assumed to be correct, thus only the elements of p(t) need to be estimated.

$$\delta_1(i) = \pi_i b_i(U(1)), \qquad Lo(1) \le i \le Up(1) \tag{4.2}$$

where $b_i(U(1))$ is the probability that state i produces the observation U(1) at time t =1.

Calculates the probability of being in state i, given $b_i(U(1))$, and the probability of initially being in state i, (π_i).

$$\psi_1(i) = \hat{s}(0), \qquad Lo(1) \le i \le Up(1) \tag{4.3}$$

Initialises the state array, which is used to backtrack the state sequence.

$$\tilde{s}(1) = \underset{Lo(1)\le i \le Up(1)}{\arg\max} [\delta_1(i)] \tag{4.4}$$

Determines the state(s) at t = 1, with the maximum value of δ.

2) Recursion:

Based on $\tilde{s}(t-1)$ obtain $r(t)$ and compute the upper and lower limits for the following calculations as follows.

$$\text{Lower Limit; } Lo(t) = [r(t), p(t) = 0]$$
$$\text{Upper Limit; } Up(t) = [r(t), p(t) = 1]$$

Note: There may have been more than one state in $\tilde{s}(t-1)$, this is due to the convolutional code's mapping of many states to few observations. Therefore the above limits (regions) need to be calculated for every state in $\tilde{s}(t-1)$ and all of the following calculations are done using all the regions obtained due to each state in $\tilde{s}(t-1)$.

$$\delta_t(j) = \max_{Lo(t-1)\le i\le Up(t-1)} \left[\delta_{t-1}(i)a_{ij}\right] b_j(U(t)), \qquad \begin{array}{l} Lo(t) \le j \le Up(t) \\ 2 \le t \le T \end{array} \tag{4.5}$$

where $b_j(U(t))$ is the probability that state j produces the observation U(t) at time t.

Calculates the probability of being in state j at time t, given $b_j(U(t))$ and the best score (highest probability path) from all valid states at time t-1 to state j at time t.

For the processing of the next block of data, when t = T - h, set $\pi_i = \delta_{T-h}(i)\ \forall$ valid regions of i.

$$\psi_t(j) = \underset{Lo(t-1)\le i\le Up(t-1)}{\arg\max} \left[\delta_{t-1}(i)a_{ij}\right], \qquad \begin{array}{l} Lo(t) \le j \le Up(t) \\ 2 \le t \le T \end{array} \tag{4.6}$$

Maintains the state which gave the best score (highest probability path) from all valid states at time t-1, to state j at time t.

$$\tilde{s}(t) = \underset{Lo(t)\le j\le Up(t)}{\arg\max}\ \delta_t(j) \tag{4.7}$$

Determines the state(s) at time t, with the maximum value of δ.

Note: This operation may also be used to include states which have probability values within some "threshold" value of the maximum value of δ. Numerical implementation in determining if probability values are equal may require this condition be included.

3) Termination:

$$\hat{s}(T) = \underset{Lo(T)\le i\le Up(T)}{\arg\max} \left[\delta_T(i)\right] \tag{4.8}$$

4) Path (state sequence) backtracking:

$$\hat{s}(t) = \psi_{t+1}(\hat{s}(t+1)), \quad t = T-1, T-2, \ldots, h. \tag{4.9}$$

For the processing of the next block of data, $\hat{s}(0)$ is set to $\hat{s}(T-h)$.

In order to prevent numerical and scaling problems, the log of the probabilities and associated additions replacing multiplications are used.

The difference between the RSSE and RCVA: is in the number of reduced states each algorithm performs the above calculations for at each time (t). The number of reduced states is a function of the number of states obtained from Eqns. (4.4) or (4.7). Each state obtained from Eqns. (4.4) or (4.7) is used to determine a region of states between a lower limit (Lo) and an upper limit (Up) and the size of this region is determined by μ in Eqn. (4.1). The RSSE version of the above algorithm must perform all of the above calculations on a fixed number of reduced states at each time t. Therefore only one state can be obtained from Eqns. (4.4) or (4.7) at each time t, and this state is used to determine the region of reduced states to be used in the calculations at the time t+1. All the other calculations and the retention of path history is still performed for all states within the reduced region. In ISI problems it is usually the case that only one state will be obtained from Eqns. (4.4) or (4.7). However when there are many states mapped to few observations, as in convolutional coded signals problems, more than one state may be obtained from Eqns. (4.4) or (4.7) at varying times during the course of processing all the data. This is especially evident at start up or after some change in environment, such as a noise burst which causes lock on the correct state sequence (path) to be lost. When lock onto the correct state sequence is lost, the probability values, δ, begin to flatten which leads to more than one state maximising δ (or being within a "threshold" value of the maximum δ). Therefore the RCVA begins increasing the range of states to be searched in order to reacquire lock onto the correct state sequence and thus the RCVA may process a large number of states for some time periods until lock onto the correct state sequence (path) is again regained. An RSSE version of the above algorithm can still be used in the case of many states mapped to few observations by selecting only one state (arbitrarily) from Eqns. (4.4) or (4.7) at each time t, and therefore maintain a fixed number of reduced states during the course of processing all the data.

When L superimposed signals are considered the system can be considered as a single convolutional coded signal having an effective constraint length LN and rate $^{LZ}\!/_{LQ}$. The state of the system's shift register can be considered to be in one of the following two forms:

$$\begin{aligned} S(t) &= \left[s^{(1)}(t), s^{(2)}(t), \ldots, s^{(L)}(t)\right] \\ &= \left[b^{(1)}(t-N+1), \ldots, b^{(1)}(t-1), b^{(1)}(t), b^{(2)}(t-N+1), \ldots, b^{(2)}(t-1),\right. \\ &\qquad \left. b^{(2)}(t), \ldots, b^{(L)}(t-N+1), \ldots, b^{(L)}(t-1), b^{(L)}(t)\right] \end{aligned} \tag{4.10}$$

where the overall state of the system's shift register is a concatenation of the states of the individual signal's shift registers, or

$$\begin{aligned} S(t) = &\left[b^{(1)}(t-N+1), b^{(2)}(t-N+1), \ldots, b^{(L)}(t-N+1), \ldots, b^{(1)}(t-1),\right. \\ &\left. b^{(2)}(t-1), \ldots, b^{(L)}(t-1), b^{(1)}(t), b^{(2)}(t), \ldots, b^{(L)}(t)\right] \end{aligned} \tag{4.11}$$

where the overall state of the system's shift register is obtained by interlacing the individual signal's shift registers.

Either form can be used if decoding is to be accomplished via the optimal full state VA. Once the system state has been estimated the individual signals' states can be determined. Reduced complexity Viterbi decoding of superimposed convolutional coded signals can be accomplished as follows.

Eqn. (4.11) is used to represent the system state, so that the bits which represent each signals' shift register state are evenly distributed in the overall system state representation, thus removing any estimation bias towards one signal, and Eqn. (4.11) can then be partitioned as follows:

$$S(t) = [R(t), P(t)] \tag{4.12}$$

where

$$R(t) = \left[b^{(1)}(t-N+1), b^{(2)}(t-N+1), \ldots, b^{(L)}(t-N+1), \ldots, \\ b^{(1)}(t-\mu-1), b^{(2)}(t-\mu-1), \ldots, b^{(L)}(t-\mu-1)\right],$$

$$P(t) = \left[b^{(1)}(t-\mu), b^{(2)}(t-\mu), \ldots, b^{(L)}(t-\mu), \ldots, b^{(1)}(t), b^{(2)}(t), \ldots, b^{(L)}(t)\right],$$

and $0 \le \mu < N$.

In the above partition, R(t) is obtained from S(t-1). Note that the number of bits in R(t) is an integer multiple, L, of the number of bits in r(t). Although the number of bits in R(t) could in fact be any number, keeping it as defined above also prevents one signal being biased over another. A full state VA would now require 2^{LN} states, while the RCVA would generally only require $2^{L(\mu+1)}$ states. The RCVA can now be used to decode the system state sequence with the appropriate replacement of s(t) = r(t)p(t) to S(t) = R(t)P(t). Estimates of the individual state sequences are then obtained from the estimated system state sequence.

§4.3 On-Line Joint State Sequence and Parameter Estimation

This section describes the reduced complexity on-line state sequence and parameter estimation algorithm for superimposed convolutional coded signals. The algorithm can be summarised as follows:

Step 1: Perform reduced complexity Viterbi decoding, using the RCVA in an on-line mode (using overlapping blocks of T observations), given estimates of the parameters of the signals. Step 1 is described in Section 4.2.

Step 2: For each time sample (in the block of T observations) update the parameter estimates using a recursive (on-line) version of the EM algorithm for superimposed deterministic signals. The parameter estimates at time $t = T$ are passed back to step one for the next stage of reduced complexity Viterbi decoding. In the rest of this section Step 2 is described.

Parameter estimation for superimposed deterministic signals can be accomplished by processing a block of data to produce one estimate for each of the parameters of interest. Feder and Weinstein's [1988] EM Algorithm for superimposed deterministic signals can be used to solve this block processing problem, as shown in Chapter 3, Section 3.3.2.

A recursive version of the EM algorithm for superimposed deterministic signals is presented below. The algorithm updates estimates of the parameters at each time sample. The recursive scheme is based on the sequential algorithms of Weinstein *et al.* [1990] and the on-line scheme of Krishnamurthy and Moore [1993]. Recursive EM algorithms have successfully been used in these papers to achieve multi-sensor signal enhancement and HMM estimation.

As described in Weinstein *et al.* [1990] and Krishnamurthy and Moore [1993] the recursive EM algorithm is a stochastic gradient method that maximises the Kullback Leibler information measure. The EM algorithm described below recursively separates the signals and updates the signals parameters by decoupling the joint log-likelihood function using knowledge of the signal models. The signal separation is similar to what is done in adaptive array processing [Haykin 1991] except that in the EM approach (below) it is accomplished using signal model information. At each time sample, t, and for each signal, ℓ, $1 \leq \ell \leq L$, calculate the following:

The E-step:

Estimate the signal component, $u_\ell(t)$ †, of the ℓ^{th} signal, $z_\ell(s_\ell(t),\Theta_\ell(t))$, by decoupling the observed data, $U(t)$ ‡. This is accomplished as follows:

$$u_\ell(t) = z_\ell(s_\ell(t),\Theta_\ell(t)) + \zeta_\ell\left[U(t) - \sum_{r=1}^{L} z_r(s_r(t),\Theta_r(t))\right] \tag{4.13}$$

where $s_\ell(t)$ is the deterministic part of the signal (in this case it is the shift register state of signal ℓ at time t),

$\Theta_\ell(t)$ is the vector of parameter estimates of signal ℓ at time t, and

the ζ_ℓ's are arbitrary real-valued scalars, that sum to 1, see Appendix D for more details about choosing its values.

The M-step:

Update each one of the ℓ^{th} signal's parameter estimates for time t+1 (see Weinstein *et al.* [1990] and Krishnamurthy and Moore [1993]) using the partial derivatives (with respect to the parameter in question $\gamma(t)$) of the log-likelihood function of $u_\ell(t)$. That is:

$$\gamma_\ell(t+1) = \gamma_\ell(t) + I_\ell(t)\left[\frac{dQ_\ell(t)}{d\gamma_\ell(t)}\right] \tag{4.14}$$

where $\gamma(t)$ represents each element of $\Theta_\ell(t)$ (i.e. the signal's parameters — amplitude, phase offset and AOA) in turn,

---

† The "complete" data, see Appendix D.
‡ The "incomplete" data, see Appendix D.

$Q_\ell(t) = \log \Pr\{u_\ell(t) | z_\ell(s_\ell(t), \Theta_\ell(t))\}$ is the log-likelihood function of $u_\ell(t)$, see Chapter 3, Section 3.3.2 for details.

$I_\ell(t)$ may be defined a variety of ways. These include using the Fisher Information Matrix or score vector of $Q_\ell(t)$, if they can be computed. $I_\ell(t)$ may also be defined so as to allow exponential forgetting which reduces the effect of past observations when estimating time varying parameters. For more detailed information on sequential algorithms the reader is referred to Weinstein *et al.* [1990], Krishnamurthy and Moore [1993] and Titterington *et al.* [1985].

§4.4 Simulations and Results

In this section the performance of the RCVA is compared to the RSSE and VA. This is done in an on-line mode. On-line processing is performed on overlapping blocks of data, so that the memory requirements of decoding the signal is reduced from the complete data length to the length of each block. The blocks are overlapping so that the initial "h" estimates of each block can be obtained from the final "h" estimates of the previous block ("h" is the delay after which a state does not affect the observations, see Section 4.2). Processing in small blocks of data also allows the determination of the time at what the algorithms acquire the correct state sequence. Next the performance of the reduced complexity on-line joint state sequence and parameter estimator is examined. This includes testing the estimator with the state sequence estimated via the RCVA, RSSE and VA, again in an on-line mode. In the simulations, two signals are incident on an ULA of 5 sensors. The signals and array details are the same as described in Chapter 3, Section 3.4.2.2. The signals are again incident on the array at 23° and 28° from endfire and the SNR/SENSOR was 20dB for each signal.

§4.4.1 Performance of the RCVA vs RSSE and VA

In these simulations, it is assumed that all parameters of the signals are known and not estimated. This allows just the performance of the three sequence estimators to be compared. Both signals are decoded simultaneously using a composite signal with constraint length 14 and rate $\frac{2}{4}$, obtained by interlacing the shift registers of each signal, as in Section 4.2, Eqn. (4.11). The constraint length of this composite signal is too large for state of the art Viterbi decoders, thus reduced state techniques would normally need to be used.

Figure 4.1 shows the results of estimating the state sequence of the composite convolutional coded signal described above, these results are the average of 100 realisations. The composite state sequence was estimated using the RCVA, RSSE and VA, all in on-line modes. For the RCVA and RSSE algorithms the reduced number of states is set by μ. In this example $\mu = 2$ for each signal was used, thus giving a reduced state region of $2^{2*(2+1)} = 2^6 = 64$ states (out of $2^{14} = 16384$) for the composite signal. The initial state estimate for all three algorithms was randomly chosen (but was the same for each method in each realisation) in a different reduced state region to the correct initial state. All algorithms were computed using their logarithmic versions in order to reduce numerical and scaling problems. The RCVA used all states whose probability value was within a "threshold" of 15.0 of the maximum probability value to determine which regions are used at the next time sample, thus for some time samples more than 64 states may be used in the computations. When more than one state maximised Eqns. (4.4) or (4.7) the RSSE version arbitrarily selected only one of these states in order to determine which reduced region to use at the next time sample, thus maintaining the search region at each time sample at 64 states. The total number of samples in the realisation was 4028, the number of samples per block, T, was 128, with an overlap of h = 28. Due to this overlap the total number of time samples used in the computation was 5159.

After the composite state sequence is estimated it is then decomposed into the state sequence of each signal and an average Bit Error Rate (BER) of both signals computed. The plot shows the average BER of the two signals for each processed block.

Figure 4.1 reveals that the VA acquired the correct sequence within the first processing block. The RCVA achieved lock (on average) after 946 samples or approximately 9 blocks of processing and all realisations acquired the correct state sequence within the 4000 time samples. The RSSE version achieved lock (on average) after 2072 samples or approximately 21 blocks of processing, however 22 or more than $\frac{1}{5}$ of the realisations did not acquire the correct state sequence within the 4000 time samples (neglecting these realisations gives an average time to acquire lock of 1528 samples).

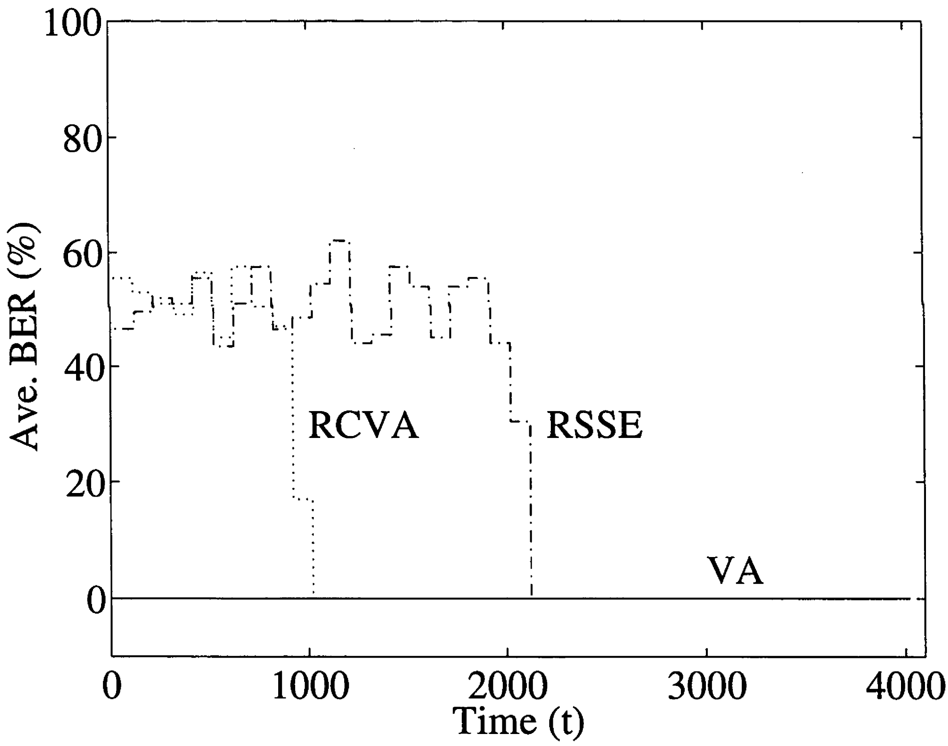

Figure 4.1: Ave. BER for VA, RCVA and RSSE ($\mu = 2$) with no parameter estimation.

Computing the probability value for each state involves a similar number of additions and comparisons for all methods, thus to compare the computational complexity of the three methods, it is sufficient to compare the total number of states searched and the maximum number of states searched at any one time, which determines the memory requirements. A Comparison of the three methods follows and also included are the figures that correspond to using a sequential (beamformer) method to separate the signals and then decode them individually using multiple ($2^7 = 128$ state) VAs (as described in Chapter 1, Section 1.4).

Method	Total No. of states searched	Max. No of states searched at any one time
VA (Full number of States)	16384*5159 = 84,525,056	16,384
Sequential (Beamformer) Method	2*128*5159 = 1,320,704	2*128 = 256
RCVA (Average for 100 realisations)	584,703	1024
RCVA (Maximum in 100 realisations)	802,624	1856
RSSE	64*5159 = 330,176	64

As can be seen, the reduced state algorithms (RCVA and RSSE) have computational complexity less than a beamforming method and are significantly less complex than using the full state VA for the composite signal. The RCVA is (on average) approximately double the computational complexity of the RSSE method, however this increased complexity improves the probability that the RCVA acquires the correct state sequence. As stated earlier the RSSE method did not acquire the correct state sequence in more than $1/5$ of the realisations, this is due to its arbitrary choice of state when more than one state has the same maximum probability of occurring. Also the maximum number of states searched by the RCVA is substantially less than the number of states search by full state VA.

These simulations illustrate that the RCVA performs (on average) significantly better than the RSSE while only requiring approximately double the computational complexity. The RCVA also performs only a little worse than the full state VA, but its computational complexity is substantially less.

§4.4.2 On-Line Joint Estimator

The two signals (described at the beginning of Section 4.4) are too closely spaced to apply beamformer techniques to separate the signals, which would allow successful individual decoding of each signal (see performance of the ML method in Chapter 3, Section 3.4.2.2). Simulations are now used to compared the performance of the RSSE, RCVA and VA methods when they are used to estimate the signals' state sequences as part of the on-line joint estimator. The parameter estimation (for each method) is achieved using the on-line EM algorithm for superimposed deterministic signals described in Section 4.3. The signals' amplitude and phase offset are assumed known and not estimated and the initial state estimate for all three methods was again randomly chosen in a different reduced state region to the correct initial state (but was the same for each method in each realisation). Each plot is of a realisation which is close to the average result of 100 realisations and show the average BER of the two signals for each processed block as well as the estimate of the AOAs of each signal at each time sample. The initial AOA estimates were 1° and 50°. After 2000 samples the true AOAs of the two signals were each changed by 5° (from 23° and 28° to 28° and 33° respectively) and thus exponential forgetting was incorporated into the parameter estimation by approximating $I_\ell(t)$, in Eqn. (4.14), as follows:

$$I_\ell(t) = \left[\frac{1-a^t}{1-a} + 100\right]^{-1}$$

"a" determines the rate of forgetting and in these simulations was set to 0.95. The "+ 100" is required to prevent numerical problems while t is small.

Figure 4.2 shows an average realisation which used the RSSE ($\mu = 2$) for estimating the state sequence. The average number of samples to acquire lock onto the correct state sequence was 3393, which included 34 realisations (or more than ⅓) that did not acquire the correct state sequence within the 4000 samples (neglecting these realisations gives an average number of 3333 samples to acquire the correct state sequence). All realisations (even the ones which didn't acquire the correct state sequence) showed that the AOA estimates stayed approximately correct, but still varied around the correct values while the average BER of the signals remained approximately 50%. This is until the correct state sequence is acquired, when the BER goes to zero and the AOA estimates improve (note the last 800 samples on this plot). The fact that the AOA estimates stay approximately correct when the BER is still ~50% is due to the fact that ¼ of all states in each convolution coded signal produces the same symbol, thus while the state estimate maybe incorrect (and hence the BER ~50%), for a good number of the estimates the symbol used in the parameter estimation will be the correct one, thus allowing the AOA estimates to remain approximately correct.

Figure 4.3 shows an average realisation which used the RCVA ($\mu = 2$) for estimating the state sequence. The average number of samples to acquire lock onto the correct state sequence was 1495, which included 2 realisations that did not acquire the correct state sequence within the 4000 samples (neglecting these 2 realisations gives an average number of 1444 samples to acquire the correct state sequence). The plot shows that the algorithm tracked the change in AOAs.

Figure 4.4 shows an average realisation which used the full state RCVA (i.e. $\mu = 7$, which is therefore an on-line VA) for estimating the state sequence. This plot shows that lock onto the correct state sequence and AOA estimates was achieved (on average) after 345 samples and that the algorithm also tracked the change in AOAs.

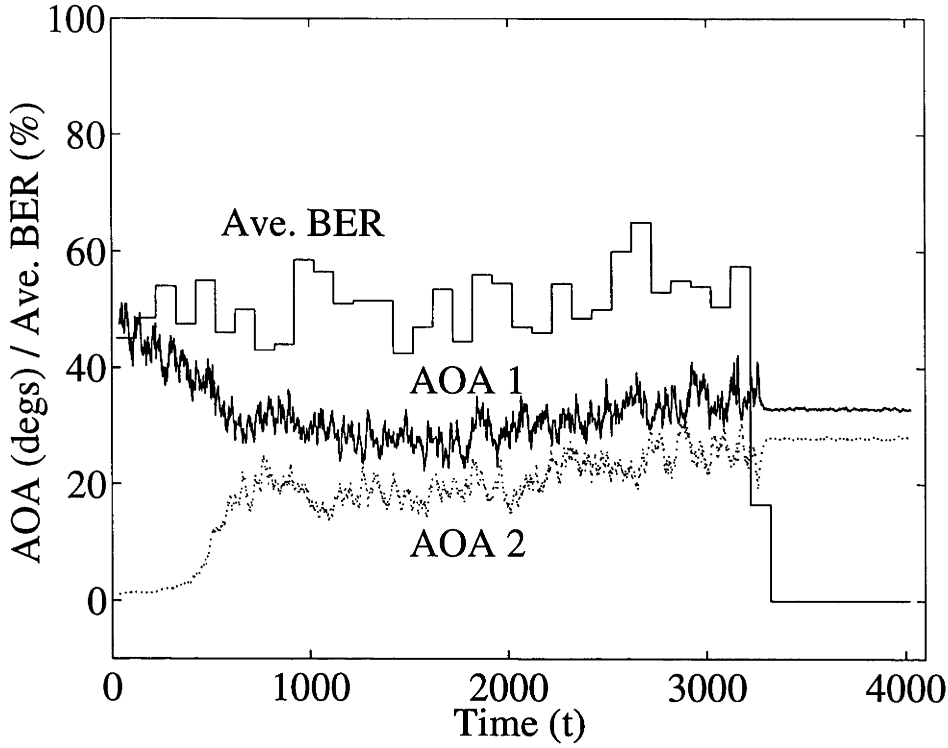

Figure 4.2: Ave. BER and AOA estimates using RSSE (μ = 2).

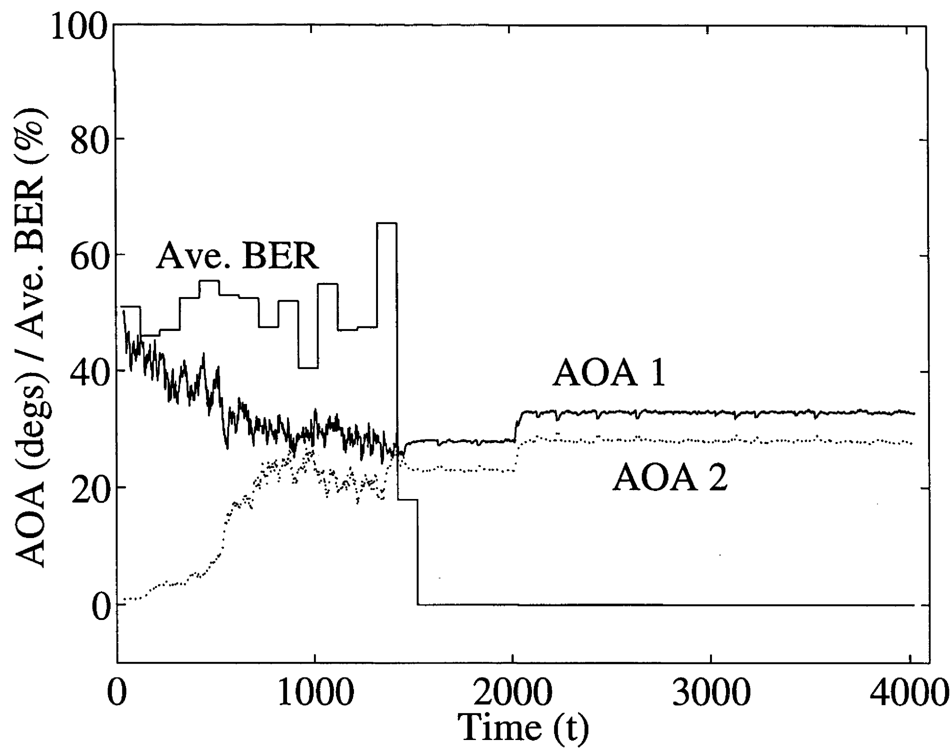

Figure 4.3: Ave. BER and AOA estimates using RCVA (μ = 2).

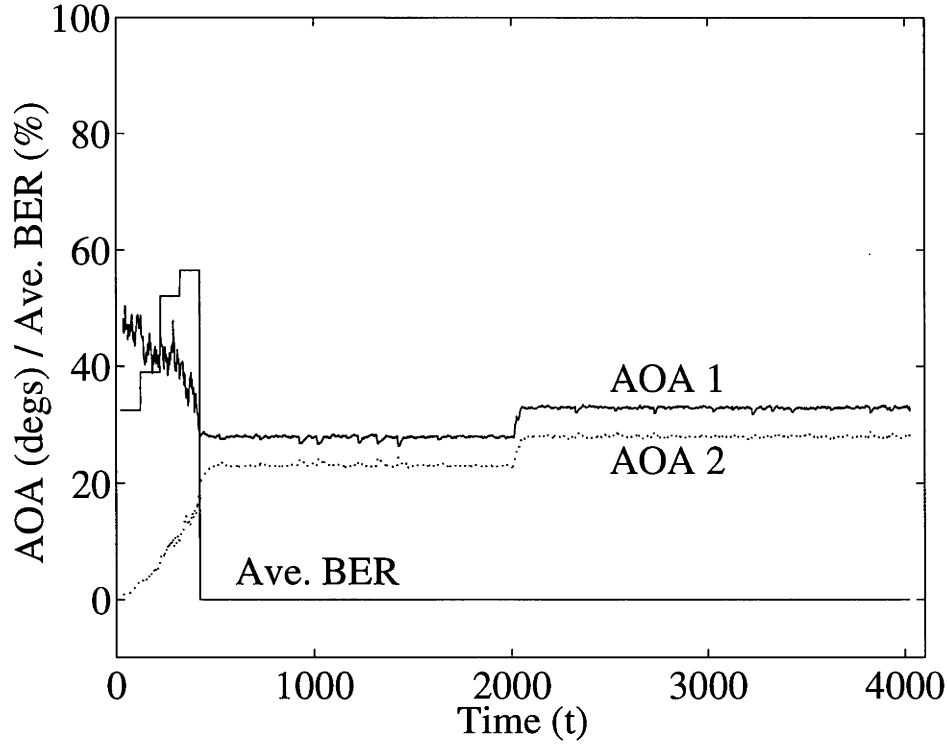

Figure 4.4: Ave. BER and AOA estimates using an on-line VA.

Again the computational complexity of the methods are compared in the following table. The parameter estimation step has the same computational complexity for all methods and therefore is not included in the table, and the only results which have changed (for the sequence estimation step) from the previous comparison (in Section 4.4.1) is for the RCVA method (the other results are repeated to ease comparison). Again the results illustrate that the RCVA performs (on average) significantly better than the RSSE while only requiring (on average) approximately double the computational complexity. The RCVA also performs only a little worse than the full state VA, but its computational complexity is substantially less.

Method	Total No. of states searched	Max. No of states searched at any one time
VA (Full number of States)	16384*5159 = 84,525,056	16,384
Sequential (Beamformer) Method	2*128*5159 = 1,320,704	2*128 = 256
RCVA (Average for 100 realisations)	633,899	1256
RCVA (Maximum in 100 realisations)	1,028,288	3520
RSSE	64*5159 = 330,176	64

§4.5 Conclusion

In this chapter a method which decreases the computational complexity of jointly estimating the state sequences and parameters of superimposed convolutional coded signals was developed. A reduced complexity Viterbi algorithm was developed for estimating superimposed signals. The RCVA differs from standard RSSE methods by allowing the number of reduced states to be varied rather than remain fixed. The joint estimation was achieved by iteratively estimating the state sequences (using the RCVA) and estimating the signals' parameters (using an on-line EM algorithm for superimposed deterministic signals). Simulations illustrate the significantly improved performance (on average) of the RCVA over the RSSE in acquiring lock onto the correct state sequences and parameter estimates, while only performing a little worst than if the VA was used. The RCVA (on average) only requires approximately double the computational complexity of the RSSE method, but its computational complexity is substantially less than using the VA. The on-line EM method, described in Section 4.3, produces an AOA estimate directly from the solution of an equation rather than having to be obtained via grid searches (as in Chapter 3), thus further reducing the computational complexity.

Chapter 5

A Hybrid Viterbi / HMM Algorithm

§5.1 Introduction

In Chapter 1, it was stated that the VA is used in most digital telecommunications applications. A drawback of the VA is that it is difficult to propagate confidence of the ML state sequence, due to the backtracking procedure which produces hard decision state estimates. These hard decision state estimates can cause a propagation of errors when two VAs are used in a concatenated way, e.g. in joint demodulation, decoding and equalisation problems [Hagenauer and Hoeher 1989]. As a consequence there has been considerable interest in soft-output VAs, a survey of these is provided by Vucetic and Li [1994].

The HFBA computes an *a posteriori* probability for each state at each time given the set of observations, thereby providing a measure of uncertainty (or confidence) for each state estimate. However, apart from the computational cost of this algorithm (an issue which is becoming less important in many applications as fast DSP chips become more affordable) the main criticism of the state estimates generated is that there is no respect for the path constraints of the model. In some signal processing applications where only a few state transitions are valid, this is seen as a severe disadvantage (e.g. in convolutional decoding, using the HFBA can produce an estimated state sequence with invalid path transitions).

At present either the VA or the HFBA is used in applications (e.g. signal processing and speech). In order to compute the ML state sequence estimate or the MAP state estimate, different computer programs or subroutines or hardware needs to be used.

In Section 5.2, a Viterbi forward-backward algorithm (VFBA) is derived from the classical VA. This is done by presenting the VA's path metric as a forward path probability and in an analogous manner deriving a backward path probability. Combining these probabilities gives a VFBA which computes an *a posteriori* probability for a given state maximised over all state sequences passing through that state. This new probability measure will now be referred to as the "*a posteriori* maximum path probability" (AMPP). Observe that a probability is obtained for each state at a given time, thus yielding uncertainty information (or confidence levels) about each state estimate. The AMPP values could be used as soft outputs in the inner decoding block of communication systems. It is demonstrated that by choosing the state estimate which has the maximum AMPP at each time, produces a state sequence which is the same as would be obtained via the VA.

It is noted that the VFBA has a structure closely related to the HFBA and in Section 5.3 a hybrid Viterbi / HMM forward-backward algorithm (hybrid algorithm) is derived. The hybrid algorithm is based on the Varadhan-Laplace lemma [Baras and James, Friedlin and Wentzell 1984]. This hybrid algorithm interpolates between the VFBA and the HFBA, thereby providing a method for obtaining varying degrees of reliance on path constraints. The hybrid algorithm depends on a tuning parameter, $0 < \mu < \infty$, and in the limits, $\mu \to \infty$ and $\mu \to 0$, it is shown that the hybrid algorithm yields the VFBA and the HFBA respectively. The use of the hybrid algorithm, via the tuning parameter μ, may also yield better adaptive algorithms.

This chapter is organised as follow: In Section 5.2 the Viterbi forward-backward algorithm is derived. The hybrid Viterbi / HMM forward-backward algorithm is derived in Section 5.3, with some conclusions presented in Section 5.4.

§5.2 A Viterbi Forward-Backward Algorithm

In this section the VFBA is derived. This algorithm computes an *a posteriori* probability for each state at each time but which is maximised over all valid paths which pass through that state. These probability values give a degree of confidence for the state estimate obtained at each time. The probability values could also be directly used as the soft outputs to a next stage VA, as required in communications systems which use concatenated VAs.

Consider the ML forward path probability $\delta_t(i)$, Eqn. (A.1), given in the VA. Assuming that one is given a block of observed data $U=\{U_1,U_2,\ldots,U_T\}$ then it is reasonable to consider a new information probability measure at time $t\in\{1,\ldots,T\}$ determined by:

$$\tilde{\gamma}_t(i)=\max_{q_1,\ldots,q_{t-1}}\max_{q_{t+1},\ldots,q_T}P[q_1,\ldots,q_{t-1},q_t=S_i,q_{t+1},\ldots q_T,U],\qquad 1\le i\le F,\quad 1\le t\le T. \tag{5.1}$$

$\tilde{\gamma}_t(i)$ is the probability of being in state S_i at time t, maximised over all possible paths which pass through state S_i. It will now be referred to as the "*a posteriori* maximum path probability (AMPP)".

To compute Eqn. (5.1), we split it into two parts (analogous to Eqn. (B.4)).

$$\begin{aligned}\tilde{\gamma}_t(i)&=\max_{q_1,\ldots,q_{t-1}}P[q_1,\ldots,q_{t-1},q_t=S_i,U_1,\ldots U_t]\max_{q_{t+1},\ldots,q_T}P[q_{t+1},\ldots q_T,U_{t+1},\ldots,U_T|q_t=S_i]\\&=\delta_t(i)\tilde{\beta}_t(i)\end{aligned} \tag{5.2}$$

Observe that the forward path probability, $\delta_t(i)$, is exactly the standard VA's ML forward path probability (see Appendix A, Eqn. (A.1)), while the backward path probability, $\tilde{\beta}_t(i)$, can be computed inductively, (analogous to Rabiner [1989]) as follows:

1) Initialisation:

$$\tilde{\beta}_T(i) = 1 \qquad 1 \leq i \leq F \tag{5.3}$$

2) Recursion:

$$\tilde{\beta}_t(i) = \max_{1 \leq j \leq N}\left[\tilde{\beta}_{t+1}(j) a_{ij} b_j(U_{t+1})\right] \qquad 1 \leq i \leq F \tag{5.4}$$

$$T - 1 \geq t \geq 1$$

The AMPP, $\tilde{\gamma}_t(i)$, is a probability value for every state at each time, based on path constraints. To obtain a state sequence estimate from the VFBA choose the state with the maximum AMPP, $\tilde{\gamma}_t(i)$, at each time t, ie.,

$$\hat{q}_t = \arg\max_i\left[\tilde{\gamma}_t(i)\right] \qquad 1 \geq t \geq T. \tag{5.5}$$

Note: That the choice of $\hat{q}_t$ (as the state with the maximum AMPP) at any time t, clearly defines only one state sequence estimate - the ML state sequence, see Eqn. (5.1).

To illustrate that the same state sequence estimate is obtained with both the VFBA (when the state with the maximum AMPP is chosen) and the VA, consider the simple four-state trellis covering 5 time units, Forney [1973, Fig. 8(a)]. The complete trellis, with each branch labelled with a length is reproduced in Figure 5.1. The state sequence estimated via the VA [Forney 1973, Fig. 8(b)] is reproduced in Figure 5.2. Figure 5.3, shows the steps of the VFBA. The solid lines show the surviving paths from the forward path probability, $\delta_t(i)$, while the dotted lines show the surviving paths from the backward path probability, $\tilde{\beta}_t(i)$. At each state the lower left number represents the path length due to $\delta_t(i)$, the lower right number, $\tilde{\beta}_t(i)$, while the top centre number represents the combined path lengths, $\tilde{\gamma}_t(i)$. The state surrounded by a dotted circle is the state with the shortest path (or maximum AMPP) and hence would be the estimated state. It is easily seen that the states surrounded with dotted circles correspond with the state sequence obtained via the VA shown in Figure 5.2.

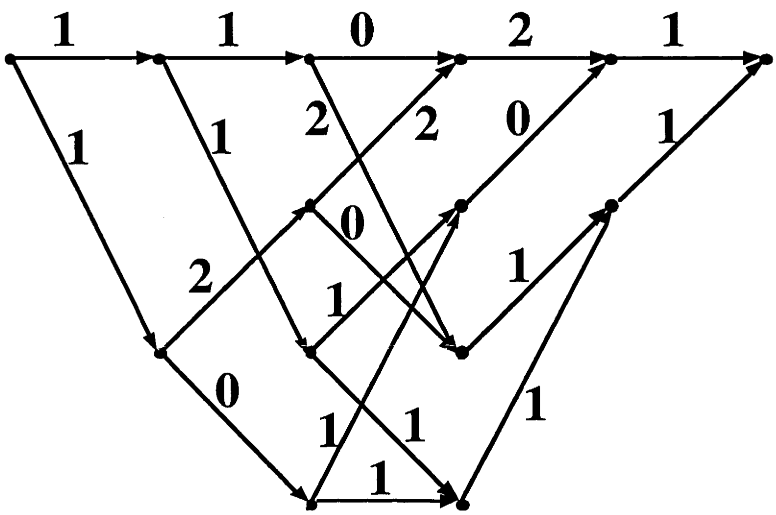

Figure 5.1: Four-State Trellis, 5 time units with branch lengths labelled.

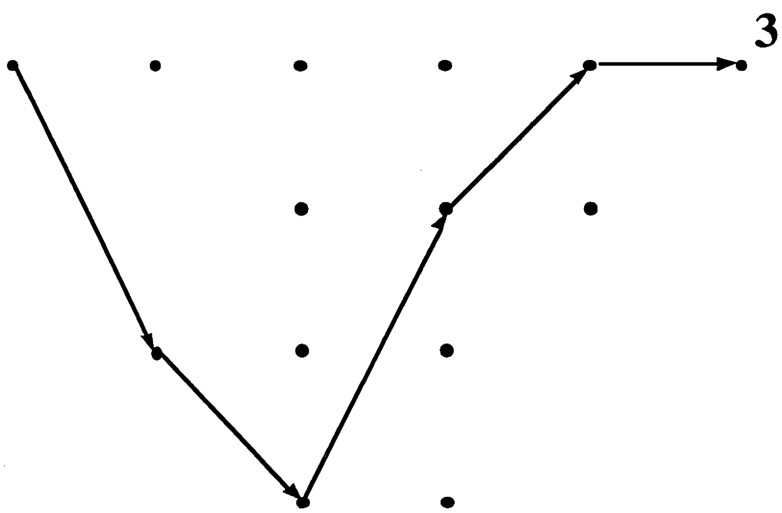

Figure 5.2: State sequence estimate obtained via the VA.

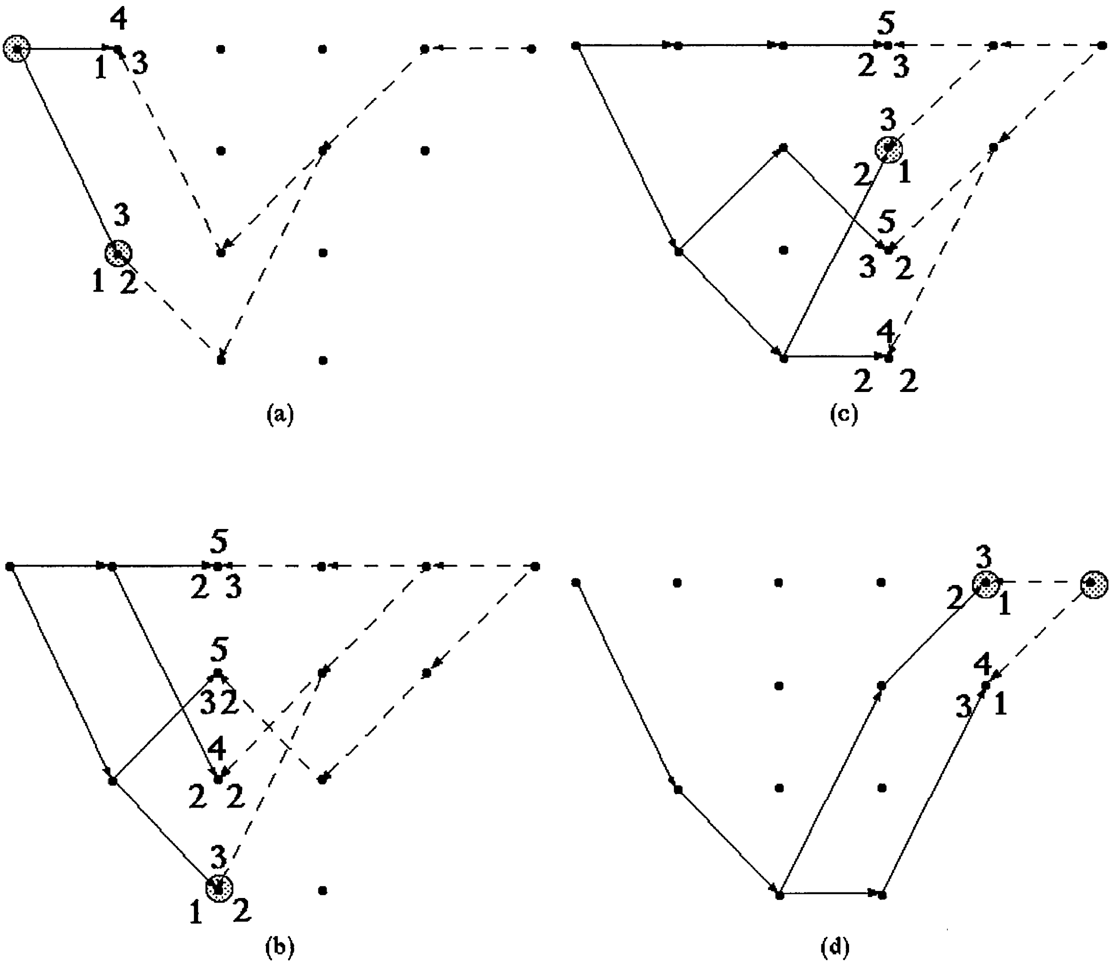

Figure 5.3: Steps to obtain state estimates using the VFBA.

Remark: It is possible to reduce the computational complexity of the VFBA by applying known HMM based techniques, (e.g. an on-line implementation using fixed-lag or sawtooth-lag smoothing [Krishnamurthy and Moore 1993]). Reduced state Viterbi techniques such as those by Eyuboğlu and Qureshi [1988], Duel-Hallen and Heegard [1989] and the RCVA developed in Chapter 4 may also be applicable.

As stated previously the AMPP, $\tilde{\gamma}_t(i)$, of the VFBA computes the probability of being in state S_i at time t maximised over all possible paths which pass through state S_i, conditioned on the observed data sequence $U = \{U_1, U_2, \ldots, U_T\}$. This compares with the VA's forward path probability, $\delta_t(i)$ (Appendix A, Eqn. A.1), which determines the probability of being in state S_i at time t maximised over all possible paths which end in state S_i, given the observed data up to time t. When t = T the two measures ($\tilde{\gamma}_t(i)$ and $\delta_t(i)$) are equivalent, however the VFBA is computationally more intensive than the VA in obtaining a state sequence estimate and therefore would only be used if there was a requirement to obtain probability measures (soft outputs) instead of actual hard-decision state estimates as obtained via the VA.

§5.3 A Hybrid Viterbi / HMM Forward-Backward Algorithm

Comparing the VFBA and the HFBA shows that the structures of these two algorithms are very similar. In this section, that similarity is exploited to develop a hybrid Viterbi / HMM forward-backward algorithm. This algorithm provides a connection, in a mathematical sense, between ML state sequence estimates (obtained via either the above VFBA or the VA) and MAP state estimates (obtained via the HFBA). The hybrid algorithm may also provide a means of relaxing the rigid path constraints imposed via ML state sequence estimation. This relaxation may be implemented by varying degrees (or adaptively) until the estimates become MAP state estimates.

Let μ be a real positive parameter (i.e. $0 < \mu < \infty$) and consider the μ dependent forward and backward probabilities ($\kappa_t^\mu(i)$ and $\tau_t^\mu(i)$ respectively) defined recursively as:

$$\kappa_t^\mu(i) = b_i(U_t)\frac{\left(1+(F-1)e^{-\mu}\right)}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\left[\exp\left(\mu\,\kappa_{t-1}^\mu(j)a_{ji}\right)\right]\right),\quad \kappa_1^\mu(i) = \pi(i), \qquad (5.6)$$

$$2 \le t \le T$$

$$\tau_t^\mu(i) = \frac{\left(1+(F-1)e^{-\mu}\right)}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\left[\exp\left(\mu\,\tau_{t+1}^\mu(j)a_{ij}b_j(U_{t+1})\right)\right]\right),\quad \tau_T^\mu(i) = 1, \qquad (5.7)$$

$$T-1 \ge t \ge 1$$

for $1 \le i \le F$. Let $\theta_t^{\mu}(i)$ be defined as the hybrid *a posteriori* probability determined by the product of the forward and backward probabilities, i.e.,

$$\theta_t^{\mu}(i) = \kappa_t^{\mu}(i)\,\tau_t^{\mu}(i) \qquad\qquad 1 \le t \le T \qquad (5.8)$$

A maximum hybrid *a posteriori* state estimate can be obtained by determining which state has the maximum value of $\theta_t^{\mu}(i)$ at each time, i.e.,

$$\hat{q}_t^{\mu} = \arg\max_{1\le j\le N}\left[\theta_t^{\mu}(j)\right], \qquad\qquad 1 \le t \le T \qquad (5.9)$$

The motivation for considering the probability measures in Eqns. (5.6) and (5.7), is given by the following proposition.

***Proposition* 1:**

Let $d_j(\mu): \Re^+ \rightarrow \Re^+$, j = 1, ..., F, be continuous functions with well defined limits at zero and infinity (i.e. $d_j^0 = \lim_{\mu\to 0}\{d_j(\mu)\}$ and $d_j^\infty = \lim_{\mu\to\infty}\{d_j(\mu)\}$). Then,

a) $$\lim_{\mu\to 0}\left\{\frac{1}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\exp(\mu\, d_j(\mu))\right)\right\} = \frac{1}{F}\sum_{j=1}^{F} d_j^0 \qquad (5.10)$$

b) $$\lim_{\mu\to\infty}\left\{\frac{1}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\exp(\mu\, d_j(\mu))\right)\right\} = \max_j\left[d_j^\infty\right] \qquad (5.11)$$

***Proof*:** Part: a)

Using the asymptotic expansion,

$$e^x = 1 + x + O(x^2) \text{ for } x \rightarrow 0,$$

where $O(x^2)$ is a function of order x^2.

gives,

$$\lim_{\mu\to 0}\left\{\frac{1}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\exp(\mu\, d_j(\mu))\right)\right\} = \lim_{\mu\to 0}\left\{\frac{1}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\left(1+\mu\, d_j(\mu)+O(\mu^2)\right)\right)\right\}$$
$$= \lim_{\mu\to 0}\left\{\frac{1}{\mu}\ln\left(1+\frac{\mu}{F}\sum_{j=1}^{F} d_j(\mu)+O(\mu^2)\right)\right\}. \tag{5.12}$$

Similarly using the approximation,

$$\ln(1+x) = x + O(x^2) \text{ for } x \to 0,$$

and observing that $\mu\sum_{j=1}^{N} d_j(\mu) = O(\mu)$ for $\mu \to 0$, gives,

$$\lim_{\mu\to 0}\left\{\frac{1}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\exp(\mu\, d_j(\mu))\right)\right\} = \lim_{\mu\to 0}\left\{\frac{1}{\mu}\left[\left(\frac{\mu}{F}\sum_{j=1}^{F} d_j(\mu)+O(\mu^2)\right)+O(\mu^2)\right]\right\}$$
$$= \lim_{\mu\to 0}\left\{\frac{1}{F}\sum_{j=1}^{F} d_j(\mu)+O(\mu)\right\}. \tag{5.13}$$

Finally taking the limit,

$$\lim_{\mu\to 0}\left\{\frac{1}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\exp(\mu\, d_j(\mu))\right)\right\} = \frac{1}{F}\sum_{j=1}^{F} d_j^0. \tag{5.14}$$

■

***Proof*:** Part: b)

Using the identity $\ln(ab) = \ln(a) + \ln(b)$ gives,

$$\lim_{\mu\to \infty}\left\{\frac{1}{\mu}\ln\left(\frac{1}{F}\right)+\frac{1}{\mu}\ln\left(\sum_{j=1}^{F}\exp(\mu\, d_j(\mu))\right)\right\}, \tag{5.15}$$

but the left hand term $\to 0$ as $\mu \to \infty$. Applying the Varadhan-Laplace Lemma to the remaining expression, yields (cf. Baras and James, Friedlin and Wentzell [1984]):

$$\lim_{\mu \to \infty}\left\{\frac{1}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\exp(\mu\, d_j(\mu))\right)\right\} = \max_j\left[d_j^{\infty}\right]. \tag{5.16}$$

■

It remains to show that this result ensures that the hybrid algorithm does in fact interpolate between ML state sequence estimation (obtained via either the above VFBA or the VA) and MAP state estimation (obtained via the HFBA). The proof of this has been split into two lemmas to aid presentation.

***Lemma* 1:**

The forward probability, $\kappa_t^{\mu}(i)$, limits to:

a) $\lim_{\mu \to 0}\{\kappa_t^{\mu}\} \to \alpha_t(i)$ The forward probability of the HFBA.

b) $\lim_{\mu \to \infty}\{\kappa_t^{\mu}\} \to \delta_t(i)$ The forward probability of the VA (and the VFBA).

***Proof* :** Part a)

Initialise $\kappa_1^0(i) := \pi(i)$. Assume that $\kappa_{t-1}^0(i) = \lim_{\mu \to 0}\{\kappa_{t-1}^{\mu}(i)\}$ exists and observe that

$$\kappa_t^0(i) := \lim_{\mu \to 0}\left\{b_i(U_t)\frac{\left(1+(F-1)e^{-\mu}\right)}{\mu}\ln\left(\frac{1}{F}\sum_{j=1}^{F}\left[\exp\left(\mu\,\kappa_{t-1}^{\mu}(j)\,a_{ji}\right)\right]\right)\right\}. \tag{5.17}$$

Taking the limit of the right hand side, noting $\lim_{\mu \to 0}\{e^{-\mu}\} = 1$, and using Proposition 1, Part a) gives:

$$\kappa_t^0(i) = b_i(U_t)\sum_{j=1}^{F}\left[\lim_{\mu\to 0}\{\kappa_{t-1}^{\mu}(j)\}a_{ji}\right] = b_i(U_t)\sum_{j=1}^{F}\left[\kappa_{t-1}^{0}(j)a_{ji}\right]. \tag{5.18}$$

Thus by induction $\kappa_t^0(i)$ exists for all t = 1, ..., T and infact $\kappa_t^0(i) = \lim_{\mu\to 0}\kappa_t^{\mu}(i)$. Comparing Eqns (5.18) and (B.5) it can be seen that,

$$\kappa_t^0(i) \equiv \alpha_t(i). \tag{5.19}$$

□

Proof **:** Part b)

Initialise $\kappa_1^{\infty}(i) := \pi(i)$. Assume that $\kappa_{t-1}^{\infty}(i) = \lim_{\mu\to\infty}\{\kappa_{t-1}^{\mu}(i)\}$ exists and using the identity $\ln(ab) = \ln(a) + \ln(b)$ observe that,

$$\kappa_t^{\infty}(i) := \lim_{\mu\to\infty}\left\{b_i(U_t)\left(1+(F-1)e^{-\mu}\right)\left[\frac{1}{\mu}\ln\left(\frac{1}{F}\right)+\frac{1}{\mu}\ln\left(\sum_{j=1}^{F}\left[\exp\left(\mu\,\kappa_{t-1}^{\mu}(j)a_{ji}\right)\right]\right)\right]\right\}. \tag{5.20}$$

Taking limit of the right hand side, noting that $\lim_{\mu\to\infty}\{e^{-\mu}\} = 0$ and using Proposition 1, Part b) gives:

$$\kappa_t^{\infty}(i) = \max_{1\le j\le F}\left[\lim_{\mu\to\infty}\{\kappa_{t-1}^{\mu}(j)\}a_{ji}\right]b_i(U_t) = \max_{1\le j\le F}\left[\kappa_{t-1}^{\infty}(j)a_{ji}\right]b_i(U_t). \tag{5.21}$$

Thus by induction $\kappa_t^{\infty}(i)$ exists for all t = 1, ..., T and infact $\kappa_t^{\infty}(i) = \lim_{\mu\to\infty}\{\kappa_t^{\mu}(i)\}$. Comparing Eqns (5.21) and (A.2) it can be seen that,

$$\kappa_t^{\infty}(i) \equiv \delta_t(i). \tag{5.22}$$

□

Lemma **2:**

The backward probability, $\tau_t^\mu(i)$, limits to:

a) $\lim_{\mu \to 0}\{\tau_t^\mu\} \to \beta_t(i)$ The backward probability of the HFBA.

b) $\lim_{\mu \to \infty}\{\tau_t^\mu\} \to \tilde{\beta}_t(i)$ The backward probability of the VFBA.

Proof **:** The proof is analogous to that for Lemma 1.

□

As a consequence of these results we show that the state estimates, $\hat{q}_t^\mu$, obtained by the hybrid algorithm limits to those obtained by the HFBA and VFBA (and thus the VA) for $\mu \to 0$ and $\mu \to \infty$ respectively.

Theorem **1:**

The hybrid algorithm interpolates between the HFBA and the VFBA. That is

$$\text{a)} \qquad \hat{q}_t^{HFBA} = \lim_{\mu \to 0}\{\hat{q}_t^\mu\}, \text{ and} \tag{5.23}$$

$$\text{b)} \qquad \hat{q}_t^{VFBA} = \lim_{\mu \to \infty}\{\hat{q}_t^\mu\}, \tag{5.24}$$

are the state estimates that would be obtained via the HFBA and the VFBA (and hence the VA) respectively.

Proof : Part a)

Taking the limit as $\mu \rightarrow 0$ of Eqn. (5.8), using Lemma 1 & 2, Part a) proves that $\theta_t^0(i) \equiv \gamma_t(i)$, and thus:

$$\hat{q}_t^0 = \lim_{\mu \rightarrow 0}\{\hat{q}_t^\mu\} \equiv \hat{q}_t^{HFBA}. \tag{5.25}$$

□

Proof : Part b)

Taking the limit as $\mu \rightarrow \infty$ of Eqn (5.8), using Lemma 1 & 2, Part b) proves that $\theta_t^\infty(i) \equiv \tilde{\gamma}_t(i)$, thus:

$$\hat{q}_t^\infty = \lim_{\mu \rightarrow \infty}\{\hat{q}_t^\mu\} \equiv \hat{q}_t^{VFBA} \equiv \hat{q}_t^{VA}. \tag{5.26}$$

□

Remark: Observe that since $\hat{q}_t^\mu$ is a hard decision from a finite set then for sufficiently small $\mu > 0$, $\hat{q}_t^\mu = \hat{q}_t^{HFBA}$ and similarly for μ sufficiently large, $\hat{q}_t^\mu = \hat{q}_t^{VFBA}$.

§5.4 Conclusion

In this chapter a Viterbi forward-backward algorithm and a hybrid Viterbi / HMM forward-backward algorithm are derived. The VFBA produces the same state estimates as the VA, but which are obtained by a maximisation of probabilities similar to the HFBA, rather than with a backtracking procedure. Therefore, the VFBA may find applications in situations which require soft output with state sequence estimation. The hybrid algorithm provides a connection, in a mathematical sense, between the maximum likelihood state sequence estimate obtained via the VFBA (and thus the VA) and the maximum *a posteriori* state estimates obtained via the classical HFBA.

Chapter 6

Conclusion

This chapter presents the dissertation conclusion. Following is an overview of the thesis and its contribution. Section 6.2 describes areas in which further investigation may be conducted, with areas for future research described in Section 6.3.

§6.1 Thesis Overview and Contribution

This thesis examined some estimation problems involving digital communications signals, in particular convolutional coded signals. It also delved into the areas of ML and MAP state sequence estimation. Technical details of the development of the contributions of this thesis were provided. The contributions are: a procedure to estimate the structure of a convolutional coded signal from only the received encoded binary data; the use of state sequence estimation combined jointly with parameter estimation in array processing problems, using knowledge of the signal models; a reduced complexity Viterbi algorithm; a reduced complexity on-line joint estimator for demodulating and estimating the parameters of superimposed convolutional coded signals, again using knowledge of the signal models; a Viterbi forward-backward algorithm; and a hybrid Viterbi / HMM forward-backward algorithm.

Chapter 1 introduced the topics and problems which were considered in this thesis. It also presented brief introductions on topics which provide background information on the research areas of interest.

Chapter 2 developed a method for determining the structure (i.e. constraint length and generating polynomials) of a rate $1/Q$ convolutional coded signal from only the received encoded data.

In Chapters 3 a method for spatial filtering superimposed convolutional coded signals using the SKMA was developed. The method differs from traditional sequential (beamformer) methods, which were described in Chapter 1, Section 1.4, in that it incorporates the signal models and jointly demodulates the signals and estimates their parameters. This method was compared with a ML estimation method. For the examples discussed, it is significantly more accurate in its demodulation of the signals and in the estimation of the AOAs, particularly at low SNR. However, this improvement is achieved through an increase in the computational complexity of the problem and hence the processing time for obtaining solutions is also increased. Also described was a means of reducing the computational complexity in the parameter estimation section of the SKMA method (using the EM algorithm for superimposed deterministic signals). This suboptimal method (SKMA - EM) significantly reduced the computational complexity involved in estimating the parameters without any significant loss of performance in the demodulation of the signals or in the estimation of the AOAs. The improved accuracy of the SKMA and SKMA - EM methods provide a remarkable threshold extension, ~20dB, compared to the ML estimation method. This improvement is due to the facts that the signals' state sequences and parameters are estimated jointly, that the discrete nature of the phase of the transmitted signals (e.g. QPSK signals) and the Markovian nature of the state sequences were included in the algorithm. In designing new antenna arrays to be used in the reception of digital communications signals (e.g. convolutional coded signals), the threshold extension obtained via the joint estimator allows smaller antenna receivers to be used, thus saving significant costs. This is however, at the expense of more powerful computing hardware (which is becoming less of a concern as computers are constantly getting smaller, more powerful and affordable).

In Chapter 4 methods for decreasing the computational complexity of jointly estimating the state sequences and parameters of superimposed convolutional coded signals were developed. In this chapter a RCVA was developed. The reduced complexity joint estimator was achieved by iteratively estimating the state sequences (via the RCVA) and estimating the signals' parameters (via an on-line EM algorithm for superimposed deterministic signals). Simulations showed the improved performance (on average) of the RCVA over the RSSE version in acquiring lock onto the correct state sequences and the parameter estimates, while only performing a little worst than if the VA was used.

Chapter 5 derived a VFBA which produces the same state sequence estimate as the VA, but which also provides confidence levels (not obtainable from the VA) for each state estimate. The VFBA has a very similar structure to the HFBA and this lead to the development of a hybrid Viterbi / HMM forward-backward algorithm. The hybrid algorithm provides a connection, in a mathematical sense, between ML state sequence estimation and MAP state estimation and also allows for the adaptive interpolation between these two methods of state estimation.

§6.2 Further Investigation

There are many different situations and minor variations in scenarios which could be envisaged to further characterise the performance of the joint estimation algorithms developed in Chapters 3 and 4. For example: complete estimation of all signal parameters, not just the AOA (remembering the amplitudes and phase offsets were assumed known and not estimated); varying the number of signals; the signals AOA separation; comparative signal strengths; environments with mixtures of interfering tones and signals; robustness against incorrect signal model assumptions (e.g. modulation type. In Chapter 3 some investigation was conducted on not modelling the generating polynomials correctly); and differing combinations of each of these, just to list a few.

Fully characterising the algorithms against all these possible situations and scenarios requires extensive simulation and time. Thus, it is be best left until the actual situation in which the algorithms are to be used is known, so that the algorithm can be completely characterised for that particular situation. A few situations may require further research in order to overcome a problem. That said, complete characterisation of the joint estimators in this thesis would probably not add much more research content, but only produce a large number of appendices containing results and figures.

For example, in completely estimating all of the signal parameters, the theory presented allows this to be done and the results from Chapter 3, Section 3.2, leads to the expectation that further simulations would show that the initial phase offset estimates would need to be at least within $\pm\pi/4$ of the correct phase offset value for each signal and that the initial amplitude estimates would not need to be all that accurate (in practical situations the initial estimates for both of these parameters would probably be obtainable by observing the signal constellation on an oscilloscope), while the initial AOA estimates would need to be within the limits determined in Chapter 3, Section 3.4.2.1. In this example no further research contribution would be obtained, only more results and figures to be placed in appendices.

In cases where the input message bits of the signals are Markovian instead of i.i.d. investigations could be conducted to determine if the use of schemes such as the "Nearly Completely Decomposable Markov Chain Scheme" by Krishnamurthy [1994] allows further reductions in the computational complexity of the Viterbi decoding instead of using the RCVA.

Further investigation into practical applications of the Viterbi forward-backward algorithm and hybrid algorithm still need to be undertaken.

§6.3 Future Research Areas

Some future research areas which could be investigated as a sequel to this thesis are now listed:

- In the estimation of the structure of a convolutional coded signal, future research could look at the case in which some bit errors occur in the received encoded data or where the signal is encoded using a punctured convolutional code [Cain *et al.* 1979]. Also in keeping with the superimposed signals problem, estimation of the structure of superimposed convolutional coded signals could be investigated.

- Removing some of the assumptions (described in Chapter 1) imposed on the signals and channel and accommodating them in the joint estimator. This would help in making the algorithms more practical. The assumptions included:

  - only additive white Gaussian noise,
  - no inter symbol interference,
  - no multipath signals,
  - signals have same baud rates and are sampled only once per baud.

- Methods of detecting when the assumed signal model (used in the joint estimators) is incorrect or some means of making the joint estimators more robust to incorrect model assumptions.

- Although estimation of the number of signals has already received research attention [Wax 1985, Wu *et al.* 1995], in keeping with the research conducted in this thesis, future research could look at estimation techniques which include knowledge of the signal models (i.e. the use of HMM information theoretic ideas [Karan 1995] could be investigated).

- Research could also be conducted into using the joint estimator with other digital communications signals (e.g. trellis coded signals [Sklar 1988]), this should just require the modification of the state estimator step of the algorithm, but other problems and complexity reduction methods may need to be researched.

- The determination of bounds (e.g. Cramer Rao Bound [Haykin 1991]) for the AOA estimates has not been considered in this thesis. Future research could study such bounds for the AOA estimates. However, this should not be mistakenly regarded as a routine task, it is a very complicated mathematical problem. This is due to the parameter estimation log-likelihood function becoming extremely complicated due to the inclusion of the signal models (see Malcolm [1995], where the signals were modelled as general linear Gauss Markov processes and he determined a CRB using an empirical technique. He stated that an analytic CRB for his system is not available).

Appendix A

The Viterbi Algorithm

Given the signal model in Chapter 1, Section 1.3, the Viterbi algorithm [Forney 1973] determines the ML state sequence over the time interval $\{1,2,\ldots,T\}$ given a sequence of observations $U=\{U_1,U_2,\ldots,U_T\}$. Omura [1969] recognised the VA to be forward dynamic programming. Let $\delta_t(i)$ denote the ML forward path probability, that is, the probability of being in state S_i at time t maximised over all possible paths which end in state S_i, with the observed data, U, up to time t:

$$\delta_t(i)=\max_{q_1,q_2,\ldots,q_{t-1}} \Pr\{q_1,\ldots,q_{t-1},q_t=S_i,U_1,U_2,\ldots,U_t\} \tag{A.1}$$

This can be computed via the recursion [Rabiner 1989]:

$$\delta_t(i)=\max_j\left[\delta_{t-1}(j)a_{ji}\right]b_i(U_t),\quad \delta_1(i)=\pi(i) \qquad 1\le i\le F \quad 2\le t\le T \tag{A.2}$$

Associated with an estimate of the most likely state at time T,

$$\hat{q}_T=\arg\max_{1\le j\le F}\left[\delta_T(j)\right] \tag{A.3}$$

there is an estimated state sequence $\hat{q}_1,\ldots,\hat{q}_{T-1}=\arg\max_{q_1,\ldots,q_{T-1}} \Pr\{q_1,\ldots,q_{T-1},q_T=\hat{q}_T,U\}$.

In practice the maximising state sequence is extracted using a backtracking process. This is done by keeping track of the argument which maximised $\delta_t(i)$ at each time,

$$\psi_t(i) = \arg\max_{1\leq j\leq F}\left[\delta_{t-1}(j)a_{ji}\right], \qquad 1 \leq i \leq F \qquad \text{(A.4)}$$

$$2 \leq t \leq T$$

and then backtracking to obtain the most likely state sequence estimate i.e.,

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \qquad t = T\text{-}1, T\text{-}2, \ldots, 1. \qquad \text{(A.5)}$$

The interested reader is referred to Forney [1973] or Rabiner [1989] for further details.

An on-line VA can be obtained in a number of ways. Two of these are:

1) After the $\delta_t(i)$s have been computed for at least "h"† observations, backtrack ("h" time periods) and output the state estimate obtained. Do this after each new calculation of the $\delta_t(i)$s. This will produce fix-lagged estimates. Saw-tooth lag estimates (see Krishnamurthy and Moore [1993]) could also be used to save having to backtrack after each new calculation of the $\delta_t(i)$s.

2) Small blocks of data which overlap by at least "h"† samples (as described in Chapter 4) could also be used to reduce the amount of backtracking required.

---

† For convolutional codes a value for "h" of 4 or 5 times the code's constraint length is sufficient [Heller and Jacobs 1971].

Appendix B

The HMM Forward-Backward Algorithm

In HMM processing, given an observation sequence and model, see Chapter 1, Section 1.3, the goal is to estimate a state sequence which is optimal is some sense. If the goal is to determine, at each separate time, the states which are individually most likely, then the MAP (or minimum variance / conditional mean) state estimates can be determined using the HFBA [Rabiner 1989].

The HFBA computes the MAP state probabilities of the transmitted sequence given a block of observations. Let $\gamma_t(i)$ denote the probability of being in state S_i at time t given the observed data sequence, $U = \{U_1, U_2, \ldots, U_T\}$:

$$\gamma_t(i) := \Pr\{q_t = S_i | U\}. \tag{B.1}$$

The MAP state estimates are obtained by maximising γ_t over i:

$$\hat{q}_t = \arg\max_i [\gamma_t(i)]. \qquad 1 \le t \le T \tag{B.2}$$

To compute γ_t, using Bayes rule, observe that

$$\Pr\{q_t = S_i | U\} = \frac{\Pr\{q_t = S_i, U\}}{\Pr\{U\}} \tag{B.3}$$

and $\Pr\{q_t = S_i, U\}$ can be factored into

$$\begin{aligned}\Pr\{q_t = S_i, U\} &= \Pr\{U_1, U_2, \ldots, U_t, q_t = S_i\} \Pr\{U_{t+1}, U_{t+2}, \ldots, U_T \mid q_t = S_i\} \\ &= \alpha_t(i)\, \beta_t(i)\end{aligned} \tag{B.4}$$

where $\alpha_t(i) = \Pr\{U_1, U_2, \ldots U_t, q_t = S_i\}$, depends only on information obtained up to time t and $\beta_t(i) = \Pr\{U_{t+1}, U_{t+2}, \ldots, U_T \mid q_t = S_i\}$, which depends only on information from t+1 to T. Commonly, $\alpha_t(i)$ and $\beta_t(i)$ are referred to as the forward and backward probabilities respectively.

They can be computed recursively as follows (cf. [Rabiner 1989]):

$$\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j)\, a_{ji}\right] b_i(U_t), \quad \alpha_0(i) = \pi(i), \qquad 1 \le i \le F, \quad 2 \le t \le T. \tag{B.5}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(U_{t+1})\, \beta_{t+1}(j), \quad \beta_T(i) = 1, \qquad 1 \le i \le F, \quad T-1 \ge t \ge 1. \tag{B.6}$$

The interested reader is referred to Rabiner [1989] for further details.

This algorithm is analogous to fixed-interval smoothing algorithms [Anderson and Moore 1979] developed in linear systems theory (e.g. the Fraser Potter smoother [1969]).

Appendix C

The Segmental k-means Algorithm

The SKMA was formally defined by Juang and Rabiner in their 1990 paper. A brief description of the algorithm follows.

The SKMA is a parameter estimation algorithm for data sequence modelling involving HMMs. The algorithm uses the state-optimised joint likelihood for the observation data and the underlaying Markovian state sequence as the objective function for estimation. The algorithm is iterative with a two step process, the segmentation and optimisation steps. In this sense the algorithm is similar to the iterative, two step, EM algorithm developed by Dempster *et al.* [1977]. Appendix D provides a brief description of the EM algorithm when applied to the problem of parameter estimation of superimposed deterministic signals [Feder and Weinstein 1988].

The segmentation step of the SKMA uses HMM methods [Rabiner 1989] (leading to the well-known VA [Viterbi 1967, Rabiner 1989]) to estimate the ML state sequences, i.e. find:

$$\hat{s} = \arg\max_{s} \Pr\{U, s|\lambda\} \tag{C.1}$$

where s is the possible states, U is the observation sequence and λ is the HMM's parameter set (see Chapter 1, Section 1.3).

Given an estimated state sequence, $\hat{s}$, and the observations, U, the optimisation step finds a new set of model parameters $\hat{\lambda}$ so as to maximise the state-optimised likelihood. That is:

$$\hat{\lambda} = \arg\max_{\lambda}\left\{\max_{s} \Pr\{U, s|\lambda\}\right\} \qquad \text{(C.2)}$$

Juang and Rabiner [1990] provide a more detailed explanation, which includes discussion on the convergence properties of the algorithm.

Appendix D

The Expectation Maximisation Algorithm for Superimposed Deterministic Signals

The now well known EM algorithm was presented in 1977 by Dempster, *et al.* This algorithm provides a method/technique for obtaining ML estimates under difficult circumstances (e.g. a likelihood function from which it may be extremely difficult or impossible to obtain ML estimates). The method maximises the conditional likelihood function of a set of *complete* (unknown) data, given observations (*incomplete*) data.

As described in Titterington *et al.* [1985], the EM algorithm has several desirable properties compared to other estimation schemes (e.g. Newton Raphson), the most important of these properties is that the likelihood function of the estimated model monotonically increases after each iteration until some convergence criteria is obtained.

Feder and Weinstein [1988] applied the EM algorithm to the problem of parameter estimation of superimposed signals. In their application they iteratively decompose the observed (*incomplete*) data into the components (*complete* data) of each signal using the expectation step and then estimate the parameters of each signal separately using the maximisation step. Using the current parameter estimates, the observed data is decomposed again in the following iteration, increasing the likelihood of the following parameter estimates. They developed their algorithm for both the deterministic (known) signals case and the stationary Gaussian signals case.

In this dissertation, the deterministic signals case is used (since estimates of the signals states are obtained from the segmentation step of SKMA, see Chapters 3 and 4) and is now detailed below:

Consider the general problem of

$$u(t) = \sum_{\ell=1}^{L} x^{(\ell)}\left(t, \Theta^{(\ell)}\right) + n(t) \tag{D.1}$$

where $\Theta^{(\ell)}$ are the vectors of unknown parameters associated with the ℓ^{th} signal component, $x^{(\ell)}\left(t, \Theta^{(\ell)}\right)$, and n(t) is the additive zero-mean WGN.

Given the observations, u(t), and assuming that the $x^{(\ell)}\left(t, \Theta^{(\ell)}\right)$ are conditionally known up to the vector $\Theta^{(\ell)}$, the goal is to jointly obtain the ML estimates of all the parameters, $\Theta^{(\ell)}$.

u(t) is referred to as the incomplete data and the complete data is defined to be the set of signal components:

$$z^{(\ell)}(t) = x^{(\ell)}\left(t, \Theta^{(\ell)}\right) + n^{(\ell)}(t). \tag{D.2}$$

where $n^{(\ell)}(t)$ are the arbitrary L decomposed components of the noise n(t), such that

$$n(t) = \sum_{\ell=1}^{L} n^{(\ell)}(t). \tag{D.3}$$

From Eqns. D.1, D.2 and D.3, the complete data, z(t), is related to the incomplete data, u(t), by:

$$u(t) = \sum_{\ell=1}^{L} z^{(\ell)}(t). \tag{D.4}$$

The EM algorithm presented by Feder and Weinstein [1988] for solving the above problem is:

The E-step:

Estimate the observation component ($\hat{z}_b^{(\ell)}(t)$) due to each signal, from the observation ($u(t)$) given the parameter estimates ($\hat{\Theta}_b^{(\ell)}$) and the deterministic signals ($x^{(\ell)}\left(t, \hat{\Theta}_b^{(\ell)}\right)$). That is:

For $\ell = 1, 2, \ldots, L$.

$$\hat{z}_b^{(\ell)}(t) = x^{(\ell)}\left(t, \hat{\Theta}_b^{(\ell)}\right) + \zeta^{(\ell)}\left[u(t) - \sum_{k=1}^{L} x^{(k)}\left(t, \hat{\Theta}_b^{(k)}\right)\right] \tag{D.5}$$

where $\zeta^{(\ell)} \geq 0$ and $\sum_{\ell=1}^{L} \zeta^{(\ell)} = 1$.

The M-step:

Obtain each signals updated parameter estimates, $\hat{\Theta}_{b+1}^{(\ell)}$, by maximising the log-likelihood functions over the Θ parameter space of each signal. That is:

For $\ell = 1, 2, \ldots, L$.

$$\hat{\Theta}_{b+1}^{(\ell)} = \arg\max_{\Theta}\left\{\log \Pr\left\{\hat{z}_b^{(\ell)}(t) \middle| \Theta\right\}\right\} \tag{D.6}$$

Note: The $\zeta^{(\ell)}$s can be used to control the rate of convergence of the algorithm and can be determined by the portion of the covariance of the complete data that can be predicted using the incomplete data [Feder *et al.* 1989]. If a prediction can not be made, the $\zeta^{(\ell)}$s are chosen to be uniform.

For a complete description of both this method and the case involving Gaussian signals, the interested reader is referred to Feder and Weinstein [1988].

Appendix E

Direction Finding via a Method of Maximum Likelihood

The method referred to as the "ML estimation method" for estimating AOAs, which is used in this dissertation is due to Wax [1985] (a description of this method can also be found in Hurt [1990]). This method is used to provide a comparative measure for the performance of the joint estimator developed in Chapter 3. A brief description of the method is now presented.

Starting with the standard baseband model of the K-vector array output, $\mathbf{U}(t) = \mathbf{A}(\Omega)\,\mathbf{Z}(t) + \mathbf{N}(t)$, Eqn. (1.11), it is assumed that: the array steering vectors (columns of $\mathbf{A}(\Omega)$) are linearly independent; the number of signals incident on the array is less than the number of sensors, K; and the noise, $\mathbf{N}(t)$, is complex valued zero-mean circular Gaussian processes with known variance, σ^2, and has stochastic i.i.d. components. Thus the joint density function of the sampled data is given by:

$$f(\mathbf{U}) = \prod_{t=1}^{T} \frac{1}{\left(\pi\sigma^2\right)^T} \exp\left(\frac{-1}{\sigma^2}\left|\mathbf{U}(t) - \mathbf{A}(\Omega)\mathbf{Z}(t)\right|^2\right). \tag{E.1}$$

The log-likelihood function of Eqn. (E.1) (ignoring constants) is given by:

$$L(\Theta) = -\frac{1}{\sigma^2}\sum_{t=1}^{T}\left|\mathbf{U}(t) - \mathbf{A}(\Omega)\mathbf{Z}(t)\right|^2 \tag{E.2}$$

where $\Theta = [\Omega(t), \mathbf{Z}(t)]$ denotes the parameter vector.

The ML estimate of the parameter vector is the value which maximises $L(\Theta)$ or equivalently minimises

$$\sum_{t=1}^{T} \left| \mathbf{U}(t) - \mathbf{A}(\Omega)\mathbf{Z}(t) \right|^2 \tag{E.3}$$

with respect to the unknown parameters (the components of Θ). This is a separable, non-linear least-squares problem which Golub and Pereyra [1973] have presented techniques for solving.

Put simply, this problem can be solved in an iterative manner by fixing one parameter as known and minimising with respect to the other parameters, substituting the solution back into Eqn. (E.3) (as a function of the fixed parameter) and then moving onto the next parameter.

Fixing Ω and minimising Eqn. (E.3), gives an estimate, $\hat{\mathbf{Z}}(t)$, i.e.,

$$\hat{\mathbf{Z}}(t) = \left(\mathbf{A}^H(\Omega)\mathbf{A}(\Omega)\right)^{-1}\mathbf{A}^H(\Omega)\mathbf{U}(t). \tag{E.4}$$

Substituting Eqn. (E.4) into Eqn. (E.3), gives:

$$\sum_{t=1}^{T} \left| \mathbf{U}(t) - \mathbf{P}(\Omega)\mathbf{U}(t) \right|^2 \tag{E.5}$$

where $\mathbf{P}(\Omega) = \mathbf{A}(\Omega)\left(\mathbf{A}^H(\Omega)\mathbf{A}(\Omega)\right)^{-1}\mathbf{A}^H(\Omega)$ is the orthogonal projection matrix spanned by the columns of $\mathbf{A}(\Omega)$, provided $\left(\mathbf{A}^H(\Omega)\mathbf{A}(\Omega)\right)^{-1}$ exists. In practice the pseudo inverse of $\left(\mathbf{A}^H(\Omega)\mathbf{A}(\Omega)\right)$ should be obtained using a singular value decomposition.

Thus the ML estimates of Ω, can be obtained from,

$$\hat{\Omega} = \arg\max_{\Omega} \left[\sum_{t=1}^{T} \left| \mathbf{P}(\Omega)\mathbf{U}(t) \right|^2 \right] \tag{E.6}$$

or alternatively,

$$\hat{\Omega} = \arg\max_{\Omega}\left[Tr\left(\mathbf{P}(\Omega)\hat{\mathbf{R}}\right)\right] \tag{E.7}$$

where $\hat{\mathbf{R}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{U}(t)\mathbf{U}^{H}(t)$ is the sample covariance matrix and *Tr* denotes the trace of a matrix.

The interested reader is referred to Wax [1985] or Hurt [1990] for more details.

The above method is only one of a number of ML based methods which can be used to estimate AOAs. The EM approach described in Appendix D, is also a ML estimation method which could be used to estimate AOAs. Miller and Fuhrmann [1990] also considered the case of ML AOA estimation via the EM algorithm for signals which are the sample path of a Gaussian process.

Bibliography

AGEE B.A., S.V. SCHELL AND W.A. GARDNER (1990). "Spectral Self-Coherence Restoral: A New Approach to Blind Adaptive Signal Extraction Using Antenna Arrays", *Proceedings of the IEEE*, vol. 78, no. 4, pp. 753 - 767.

ANDERSON J.B. AND S. MOHAN (1984). "Sequential Coding Algorithms: A Survey and Cost Analysis", *IEEE Transactions on Communications*, vol. COM-32, no.2, pp. 169 - 176.

ANDERSON B.D.O. AND J.B. MOORE (1979). *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall.

ANDERSON J.B. AND E. OFFER (1994). "Reduced-State Sequence Detection with Convolutional Codes", *IEEE Transactions on Information Theory*, vol. 40, no. 3, pp. 965 - 972.

BAHL L.R., J. COCKE, F. JELINEK AND J. RAVIV (1974). "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Transactions on Information Theory*, March, pp. 284 - 287.

BARAS J.S. AND M.R. JAMES. "Robust and Risk-Sensitive Output Feedback Control for Finite State Machines and Hidden Markov Models", submitted to *Journal Math. Sys. Est. & Cont.*

BAUM L.E. (1972). "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes", *Inequalities*, vol. 3, pp. 1 - 8.

BAUM L.E. AND J.A. EGON (1967). "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology", *Bull. Amer. Meteorol. Soc.*, vol. 73, pp. 360 - 363.

BAUM L.E. AND T. PETRIE (1966). "Statistical inference for probabilistic functions of finite state Markov chains", *Ann. Math. Stat.*, vol. 37, pp. 1554 - 1563.

BAUM L.E., T. PETRIE, G. SOULES AND N. WIESS (1970). "A Maximization technique occurring in the statistical analysis of probabilistic functions of Markov Chains", *Ann. Math. Stat.,*. vol. 41, no. 1, pp. 164 - 171.

BAUM L.E. AND G.R. SELL (1968). "Growth functions for transformations on manifolds", *Pac. J. Math.,* vol. 27, no. 2, pp. 211 - 227.

BIBB CAIN J., G.C. CLARK JR. AND J.M. GEIST (1979). "Punctured Convolutional Codes of Rate (n-1)/n and Simplified Likelihood Decoding", *IEEE Transactions on Information Theory*, vol. IT-25, no. 1, Jan., pp. 97 - 100.

BLAHUT R. E. (1984). *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley.

BURDEN R.L. AND J.D. FAIRES (1985). *Numerical Analysis* Third Edition, Boston, MA: PWS Publishers.

CAIN J.B., G.C. CLARK AND J.M. GEIST (1979). "Punctured convolutional codes of rate $(n\text{-}1)\ /\ n$ and simplified maximum likelihood decoding", *IEEE Transactions on Information Theory*, vol. IT-24, pp. 97-100.

DEMPSTER A.D., N.M. LAIRD AND D.B. RUBIN (1977). "Maximum Likelihood from Incomplete Data via the EM algorithm", *J. Roy. Stat. Soc.*, vol. B-39, pp. 1 37.

DUEL-HALLEN A. AND C. HEEGARD (1989). "Delayed Decision-Feedback Sequence Estimation", *IEEE Transactions on Communications*, vol. COM-37, no. 5, pp. 428 - 436.

DYNKIN E.B. (1965). *Markov Processes*, Volumes I and II, New York, NY: Academic Press Inc., Publisher.

ELIAS P. (1955). "Coding for noisy channels", *IRE Nat. Conv. Rec.*, vol. 3, pt. 4, pp. 37 - 46.

EYUBOĞLU M.V. AND S.U. QURESHI (1988). "Reduced-State Estimation with Set Partitioning and Decision Feedback", *IEEE Transactions on Communications*, vol. COM-36, no. 1, pp. 13 - 20.

FALCONER D.D., F. ADACHI AND G. GUDMUNDSON (1995). "Time Division Multiple Access Methods for Wireless Personal Communications", *IEEE Communications Magazine*, vol. 33, no.1, Jan., pp. 50 - 57.

FEDER M., A.V. OPPENHEIM AND E. WEINSTEIN (1989). "Maximum Likelihood noise cancellation using the EM algorithm", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 2, pp. 204 - 216.

FEDER M. AND E. WEINSTEIN (1988). "Parameter Estimation of Superimposed Signals Using the EM Algorithm", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 4, pp. 477 - 489.

FORNEY G.D., JR. (1973). "The Viterbi Algorithm", *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268 - 278.

FRASER D.C. AND J.E. POTTER (1969). "The Optimum Linear Smoother as a Combination of two Optimum Linear Filters", *IEEE Transactions on Automatic Control*, AC-14, pp. 387 - 390.

FRIEDLIN M.I. AND A.D. WENTZELL (1984). *Random Perturbations of Dynamical Systems*, New York: Springer-Verlag.

GOLUB, G.H. AND V. PEREYRA. (1973) "The Differentiation of the Pseudo Inverses and Nonlinear Least Squares Problems whos Variables Separate", *SAIM Journal of Numerical Analysis*, vol. 10, no. 2, pp. 413 - 432.

HAGENAUER J. AND P. HOEHER (1989). "A Viterbi Algorithm with Soft-Decision Outputs and its Applications", *IEEE Globecom*, Dallas, Texas, vol. 3, pp 47.1.1 - 47.1.7.

HAYKIN S. (1988). *Digital Communications*. New York: John Wiley & Sons.

HAYKIN S. (1991). *Advances in Spectrum Analysis and Array Processing,* Volume II, New Jersey: Prentice Hall.

HELLER J.A. AND I.R. JACOBS (1971). "Viterbi decoding for Satellite and Space Communications", *IEEE Transactions on Communications*, vol. COM-19, no. 5, pp. 835 - 848.

HURT N.E. (1990). "Maximum likelihood estimation and MUSIC in array signal processing : A review", *Multidimensional Systems and Signal Processing*, vol. 1, pp. 273 - 329.

INTELSAT EARTH STATION STANDARD (1987). IESS-308, REV 3, 17 JUNE.

JUANG B.H. AND L.R. RABINER (1990). "The segmented k-means algorithm for estimating the parameters of a hidden Markov model", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 9, pp. 1639 - 1641.

KARAN M. (1995). "Frequency Tracking and Hidden Markov Models", Ph.D. Dissertation, Australian National University.

KRISHNAMURTHY V. (1994). "Adaptive Estimation of Hidden Nearly Completely Decomposable Markov Chains with Applications in Blind Equalization", *International Journal on Adaptive Control and Signal Processing,* vol. 8, pp. 237 - 260.

KRISHNAMURTHY V. AND J.B. MOORE (1993). "On-Line Estimation of Hidden Markov Model Parameters Based on the Kullback-Leibler Information Measure", *IEEE Transactions on Signal Processing*, vol. 41, no. 8, pp. 2557 - 2573.

LI J. AND R.T. COMPTON, JR. (1993). "Maximum Likelihood Angle Estimation for Signals with Known Waveforms", *IEEE Transactions on Signal Processing*, vol. 41, no. 9, pp. 2850 - 2862.

MALCOLM W.P. (1995). "Algorithmic Spatial Filtering for General Linear Gauss-Markov Processes", MSc. Thesis, Flinders University of South Australia.

MALCOLM W.P. AND L.B. WHITE. "Spatial Filtering for general linear Gauss Markov Processes", in preparation.

MILLER M.I. AND D.R. FUHRMANN (1990). "Maximum-Likelihood Narrow-Band Direction Finding and the EM Algorithm", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 9, pp. 1560 - 1577.

MOULINES E., P. DUHAMEL, J.-F. CARDOSO AND S. MAYRARGUE (1994). "Subspace Methods for the Blind Identification of Multichannel FIR Filter", in *Proceedings ICASSP*, Adelaide, vol. 4, pp. 573 - 576.

OMURA J.K. (1969). "On the Viterbi Algorithm", *IEEE Transactions on Information Theory*, Jan, pp. 177 - 179

PADGETT J.E., C.G. GÜNTHER AND T. HATTORI (1989). "Overview of Wireless Personal Communications", *IEEE Communications Magazine*, vol. 33, no.1, pp. 28 - 41.

PILLAI S.U. (1989). *Array Signal Processing.* New York: Springer-Verlag.

PORITZ A.B. (1988). "Hidden Markov Models: A Guided Tour", in *Proceedings ICASSP*, New York, pp. 7 - 13.

RABINER L. R. (1989). "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 - 286.

RABINER L. R. AND B.H. JUANG (1986). "An Introduction to Hidden Markov Models", *IEEE Acoustics, Speech and Signal Processing Magazine*, Jan., pp. 4 - 16.

SCHMIDT R.O. (1986). "Multiple Emitter Location and Signal Parameter Estimation", *IEEE Transactions on Antennas and Propagation*, vol. AP-34, no. 3, pp. 276 - 283.

SHEEN W.H. AND G.L. STÜBER (1992). "Error probability for reduced-state sequence estimation", *IEEE Journal on Selected Areas in Communications,* vol. 10, no. 3, pp. 571 - 578.

SHEEN W.H. AND G.L. STÜBER (1993). "Error Probability of Reduced-State Sequence Estimation for Trellis-Coded Modulation on Interference Channels", *IEEE Transactions on Communications,* vol. COM-41, no. 9, pp. 1265 - 1269.

SKLAR B. (1988). *Digital Communications, Fundamentals and Applications.* Englewood Cliffs, NJ: Prentice-Hall.

STEELE R. (1992). *Mobile Radio Communications.* London: Pentech Press Ltd.

STOICA P. AND A. NEHORAI (1989). "Music, maximum likelihood and the Cramer-Rao bound", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 5, pp. 720 - 741.

TALWAR S., M. VIBERG AND A. PAULRAJ (1994). "Blind Estimation of Multiple Co-Channel Digital Signals Using an Antenna Array", *IEEE Signal Processing Letters*, vol. 1, no. 2, pp. 29 - 31.

TAO C. (1992). "A Generalization of Discrete Hidden Markov Model and of Viterbi Algorithm", *Pattern Recognition*, vol. 25, no. 11, pp. 1381 - 1387.

TITTERINGTON D.M., A.F.M. SMITH AND U.E. MAKOV (1985). *Statistical Analysis of Finite Mixture Distributions*, John Wiley and Sons.

TONG L., G. XU AND T. KAILATH (1991). "A new approach to blind identification and equalization of multipath channels", in *Proceedings 25th Asilomar Conference,* Pacific Grove, CA, pp. 856 - 860.

TONG L., G. XU AND T. KAILATH (1993). "Fast blind equalization via antenna arrays", in *Proceedings ICASSP*, vol. 4, pp. 272 - 275.

TRUDINGER P.L. AND L.B. WHITE (1994). "Direction of Arrival Estimation with Markov Signal Sources", in *Proceedings Second International Symposium on DSP for Communication Systems*, Adelaide, pp. 3.5.1 - 3.5.14.

VITERBI A. J. (1967). "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Transactions on Information Theory*, vol. IT-13, no.2, pp. 260 - 269.

VUCETIC B. AND Y. LI (1994). "A Survey of Soft-Output Algorithms", in *Proceedings International Symposium on Information Theory and its Applications*, Sydney, Australia, pp. 863 - 867.

WAX M. (1985). “Detection and Estimation of Superimposed Signals”, Ph.D. Dissertation, Stanford University, CA.

WEINSTEIN E., M. FEDER AND A. V. OPPENHEIM (1990). “Sequential Algorithms for Parameter Estimation Based on the Kullback- Leibler Information Measure”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 9, pp. 1652 - 1654.

WU H.-T., J.-F. YANG AND F.-K. CHEN (1995). “Source Number Estimators Using Transformed Gerschgorin Radii”, *IEEE Transactions on Signal Processing*, vol. 43, no.6, pp. 1325 - 1333.

XU G., L. TONG AND H. LIU (1994). “A New Algorithm for Fast Blind Equalisation of Wireless Communication Channels”, in *Proc. ICASSP*, Adelaide, vol. 4, pp. 589 - 592.

ZISKIND I. AND D. HERTZ (1993). “Maximum-Likelihood Localization of Narrow-Band Autoregressive Sources via the EM Algorithm”, in *Proceedings ICASSP*, Minneapolis, vol. 4, pp. 9 - 12.